UNESCO

United Nations Educational, Scientific and Cultural Organization

The Moodle Course Management System
A Free, Open Source System for Online Learning

Moodle *is a real gift to forward thinking educators*[1].

Cairo
January 2004

For more information, please send inquiries to
ci@unesco-cairo.org

---

[1]This document is compiled and updated by Dr. Tarek G. Shawki, Regional CI Advisor for Arab States. For further information, please e-mail the regional advisor at "tshawki@unesco-cairo.org"

Contents

1   What is Moodle?

Moodle is a course management system (CMS) - a software package designed to help educators create quality online courses. Such e-learning systems are sometimes also called Learning Management Systems (LMS) or Virtual Learning Environments (VLE). One of the main advantages of **Moodle** over other systems is a strong grounding in social constructionist pedagogy.

Moodle is Open Source software, which means you are free to download it, use it, modify it and even distribute it (under the terms of the GNU General Public License). **Moodle** runs without modification on Unix, Linux, Windows, Mac OS X, Netware and any other system that supports PHP, including most web host providers. Data is stored in a single database: MySQL and PostgreSQL are best supported, but it can also be used with Oracle, Access, Interbase, ODBC and others.
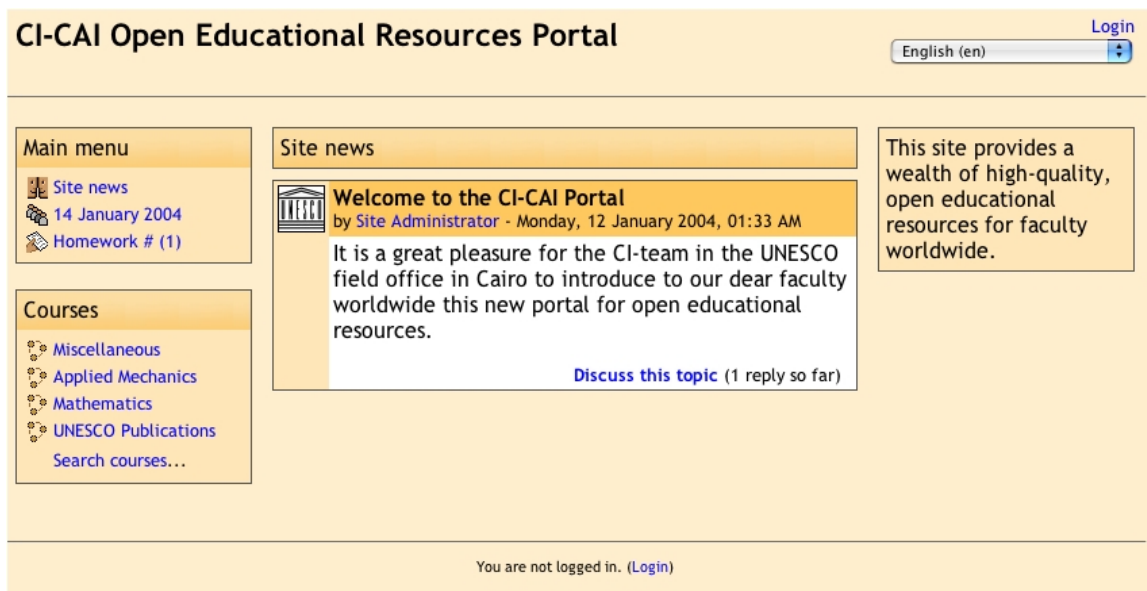
Moodle is available in 34 languages, including: Arabic, Catalan, Chinese (simplified and traditional), Czech, Danish, Dutch, English (UK and US versions), Finnish, French (France and Canada versions), German, Greek, Hungarian, Indonesian, Italian, Japanese, Norwegian, Polish, Portuguese (Portugal and Brazil), Romanian, Russian, Slovak, Spanish (Spain, Mexico, Argentina and Caribbean versions), Swedish, Thai and Turkish.

Moodle is a software package for producing internet-based courses and web sites. It's an ongoing development project designed to support a social constructionist framework of education.

The word **Moodle** was originally an acronym for Modular Object-Oriented Dynamic Learning Environment, which is mostly useful to programmers and education theorists. It's also a verb that describes the process of lazily meandering through something, doing things as it occurs to you to do them, an enjoyable tinkering that often leads to insight and creativity. As such it applies both to the way **Moodle** was developed, and to the way a student or teacher might approach studying or teaching an online course. Anyone who uses **Moodle** is a *Moodler*.

2 Features of Moodle

Moodle is an active and evolving product. This page lists just some of the many features it contains along with functionality provided by the Add-on modules:

2.1 Overall design

- □ Promotes a social constructionist pedagogy (collaboration, activities, critical reflection, etc)

- □ Suitable for 100% online classes as well as supplementing face-to-face learning

- □ Simple, lightweight, efficient, compatible, low-tech browser interface

- □ Easy to install on almost any platform that supports PHP. Requires only one database (and can share it).

- □ Full database abstraction supports all major brands of database (except for initial table definition)

- □ Course listing shows descriptions for every course on the server, including accessibility to guests.

- □ Courses can be categorised and searched - one Moodle site can support thousands of courses

- □ Emphasis on strong security throughout. Forms are all checked, data validated, cookies encrypted etc

- □ Most text entry areas (resources, forum postings, journal entries etc) can be edited using an embedded WYSIWYG HTML editor

2.2 Site management

- □ Site is managed by an admin user, defined during setup

- □ Plug-in "themes" allow the admin to customise the site colours, fonts, layout etc to suit local needs

- □ Plug-in activity modules can be added to existing Moodle installations

- □ Plug-in language packs allow full localisation to any language. These can be edited using a built-in web-based editor. Currently there are language packs for over 34 languages.

- □ The code is clearly-written PHP under a GPL license - easy to modify to suit your needs

2.3   User management

☐ Goals are to reduce admin involvement to a minimum, while retaining high security

☐ Supports a range of authentication mechanisms through plug-in authentication modules, allowing easy integration with existing systems.

☐ Standard email method: students can create their own login accounts. Email addresses are verified by confirmation.

☐ LDAP method: account logins can be checked against an LDAP server. Admin can specify which fields to use.

☐ IMAP, POP3, NNTP: account logins are checked against a mail or news server. SSL, certificates and TLS are supported.

☐ External database: any database containing at least two fields can be used as an external authentication source.

☐ Each person requires only one account for the whole server - each account can have different access

☐ An admin account controls the creation of courses and creates teachers by assigning users to courses

☐ A course creator account is only allowed to create courses and teach in them

☐ Teachers may have editing privileges removed so that they can't modify the course (eg for part-time tutors)
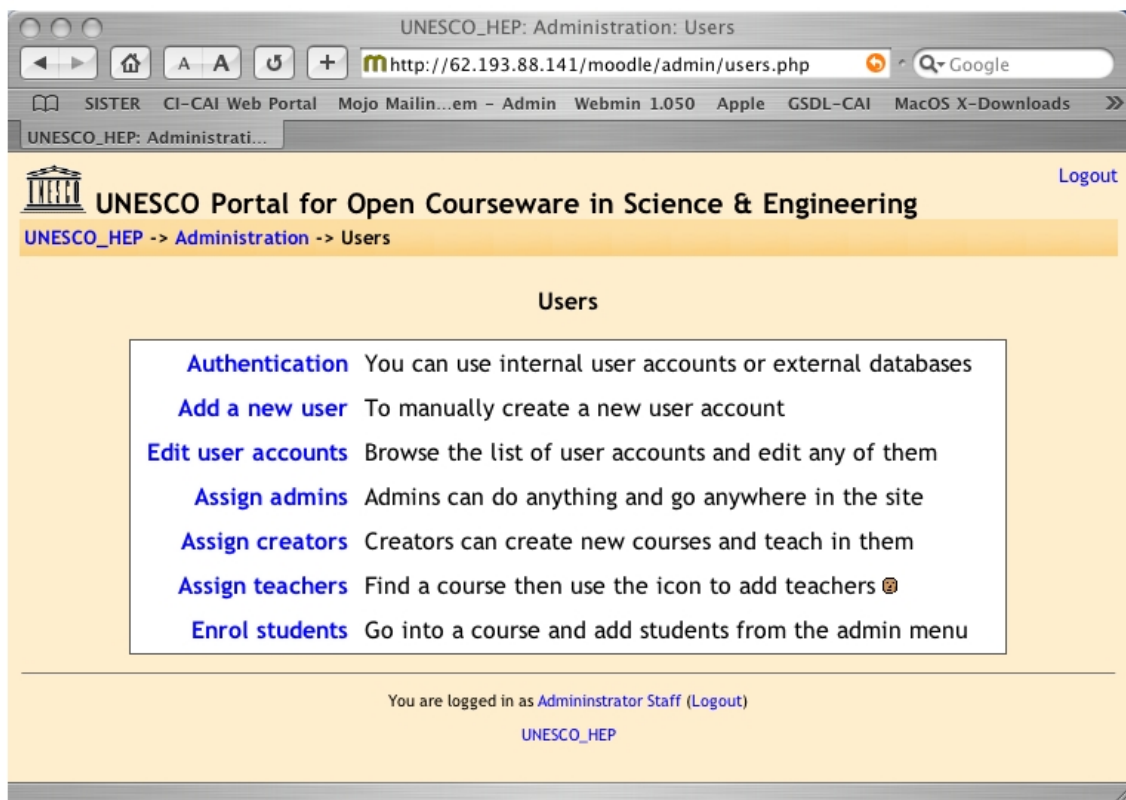
☐ Security - teachers can add an "enrolment key" to their courses to keep out non-students. They can give out this key face-to-face or via personal email etc

☐ Teachers can enrol students manually if desired

☐ Teachers can unenrol students manually if desired, otherwise they are automatically unenrolled after a certain period of inactivity (set by the admin)

□ Students are encouraged to build an online profile including photos, description. Email addresses can be protected from display if required.

□ Every user can specify their own timezone, and every date in Moodle is translated to that timezone (eg posting dates, assignment due dates etc)

□ Every user can choose the language used for the Moodle interface (English, French, German, Spanish, Portuguese etc)

2.4   Course management

□ A full teacher has full control over all settings for a course, including restricting other teachers

□ Choice of course formats such as by week, by topic or a discussion-focussed social format

□ Flexible array of course activities - Forums, Journals, Quizzes, Resources, Choices, Surveys, Assignments, Chats, Workshops

□ Recent changes to the course since the last login can be displayed on the course home page - helps give sense of community

☐ Most text entry areas (resources, forum postings, journal entries etc) can be edited using an embedded WYSIWYG HTML editor

☐ All grades for Forums, Journals, Quizzes and Assignments can be viewed on one page (and downloaded as a spreadsheet file)

☐ Full user logging and tracking - activity reports for each student are available with graphs and details about each module (last access, number of times read) as well as a detailed "story" of each students involvement including postings, journal entries etc on one page.

☐ Mail integration - copies of forum posts, teacher feedback etc can be mailed in HTML or plain text.

☐ Custom scales - teachers can define their own scales to be used for grading forums, assignments and journals

☐ Courses can be packaged as a single zip file using the Backup function. These can be restored on any Moodle server.

2.5   Assignment Module

☐ Assignments can be specified with a due date and a maximum grade.

☐ Students can upload their assignments (any file format) to the server - they are date-stamped.

☐ Late assignments are allowed, but the amount of lateness is shown clearly to the teacher

☐ For each particular assignment, the whole class can be assessed (grade and comment) on one page in one form.

☐ Teacher feedback is appended to the assignment page for each student, and notification is mailed out.

☐ The teacher can choose to allow resubmission of assignments after grading (for regrading)

2.6   Chat Module

☐ Allows smooth, synchronous text interaction

☐ Includes profile pictures in the chat window

☐ Supports URLs, smilies, embedded HTML, images etc

☐ All sessions are logged for later viewing, and these can also be made available to students

2.7   Choice Module

☐ Like a poll. Can either be used to vote on something, or to get feedback from every student (eg research consent)

☐ Teacher sees intuitive table view of who chose what

☐ Students can optionally be allowed to see an up-to-date graph of results

2.8   Forum Module

☐ Different types of forums are available, such as teacher-only, course news, open-to-all, and one-thread-per-user.

☐ All postings have the authors photo attached.

☐ Discussions can be viewed nested, flat or threaded, oldest or newest first.

☐ Individual forums can be subscribed to by each person so that copies are forwarded via email, or the teacher can force subscription for all

☐ The teacher can choose not to allow replies (eg for an announcements-only forum)

☐ Discussion threads can be easily moved between forums by the teacher

☐ Attached images are shown inline

☐ If forum ratings are being used, these can be restricted to a range of dates

2.9   Journal Module

☐ Journals are private between student and teacher.

☐ Each journal entry can be directed by an open question.

☐ For each particular journal entry, the whole class can be assessed on one page in one form

☐ Teacher feedback is appended to the journal entry page, and notification is mailed out.

2.10   Quiz Module

☐ Teachers can define a database of questions for re-use in different quizzes

☐ Questions can be stored in categories for easy access, and these categories can be "published" to make them accessible from any course on the site.

☐ Quizzes are automatically graded, and can be re-graded if questions are modified

□ Quizzes can have a limited time window outside of which they are not available

□ At the teacher's option, quizzes can be attempted multiple times, and can show feedback and/or correct answers

□ Quiz questions and quiz answers can be shuffled (randomised) to reduce cheating

□ Questions allow HTML and images

□ Questions can be imported from external text files

□ Quizzes can be attempted multiple times, if desired

□ Attempts can be cumulative, if desired, and finished over several sessions

□ Multiple-choice questions supporting single or multiple answers

□ Short Answer questions (words or phrases)

□ True-False questions

□ Matching questions

□ Random questions

□ Numerical questions (with allowable ranges)

□ Embedded-answer questions (cloze style) with answers within passages of text

□ Embedded descriptive text and graphics

2.11   Resource Module

□ Supports display of any electronic content, Word, Powerpoint, Flash, Video, Sounds etc

□ Files can be uploaded and managed on the server, or created on the fly using web forms (text or HTML)

□ External content on the web can be linked to or seamlessly included within the course interface.

□ External web applications can be linked in with data passed to them

2.12   Survey Module

☐ Built-in surveys (COLLES, ATTLS) have been proven as instruments for analysing online classes

☐ Online survey reports always available, including many graphs. Data is downloadable as an Excel spreadsheet or CSV text file.

☐ Survey interface prevents partly-finished surveys.

☐ Feedback is provided to the student of their results compared to the class averages

2.13   Workshop Module

☐ Allows peer assessement of documents, and the teacher can manage and grade the assessment.

☐ Supports a wide range of possible grading scales

☐ Teacher can provide sample documents for students to practice grading

☐ Very flexible with many options.

3   Moodle Philosophy

The design and development of Moodle is guided by a particular philosophy of learning, a way of thinking that you may see referred to in shorthand as a "social constructionist pedagogy". (Some of you scientists may already be thinking "soft education mumbo" jumbo and reaching for your mouse, but please read on - this is useful for every subject area!)

This page tries to explain in simple terms what that phrase means by unpacking four main concepts behind it. Note that each of these is summarising one view of an immense amount of diverse research so these definitions may seem thin if you have read about these before.

If these concepts are completely new to you then it is likely that these ideas will be hard to understand at first - all I can recommend is that you read it carefully, while thinking about your own experiences of trying to learn something.

3.1   Constructivism

This point of view maintains that people actively **construct** new knowledge as they interact with their environment.

Everything you read, see, hear, feel, and touch is tested against your prior knowledge and if it is viable within your mental world, may form new knowledge you carry with you. Knowledge is strengthened if you can use it successfully in your wider environment. You are not just a memory bank passively absorbing information, nor can knowledge be "transmitted" to you just by reading something or listening to someone.

This is not to say you can't learn anything from reading a web page or watching a lecture, obviously you can, it's just pointing out that there is more interpretation going on than a transfer of information from one brain to another.

3.2   Constructionism

Constructionism asserts that learning is particularly effective when constructing something for others to experience. This can be anything from a spoken sentence or an internet posting, to more complex artifacts like a painting, a house or a software package.

For example, you might read this page several times and still forget it by tomorrow - but if you were to try and explain these ideas to someone else in your own words, or produce a slideshow that explained these concepts, then I can guarantee you'd have a better understanding that is more integrated into your own ideas. This is why people take notes during lectures, even if they never read the notes again.

3.3   Social Constructivism

This extends the above ideas into a social group constructing things for one another, collaboratively creating a small culture of shared artifacts with shared meanings. When one is immersed within a culture like this, one is learning all the time about how to be a part of that culture, on many levels.

A very simple example is an object like a cup. The object can be used for many things, but its shape does suggest some "knowledge" about carrying liquids. A more complex example is an online course - not only do the "shapes" of the software tools indicate certain things about the way online courses should work, but the activities and texts produced within the group as a whole will help shape how each person behaves within that group.

3.4   Connected and Separate

This idea looks deeper into the motivations of individuals within a discussion. Separate behaviour is when someone tries to remain 'objective' and 'factual', and tends to defend their own ideas using logic to find holes in their opponent's ideas. Connected behaviour is a more empathic approach that accepts subjectivity, trying to listen and ask questions in an effort to understand the other point of view. Constructedbehaviour is when a person is sensitive to both of these approaches and is able to choose either of them as appropriate to the current situation.

In general, a healthy amount of connected behaviour within a learning community is a very powerful stimulant for learning, not only bringing people closer together but promoting deeper reflection and re-examination of their existing beliefs.

Once you are thinking about all these issues, it helps you to focus on the experiences that would be best for learning from the learner's point of view, rather than just publishing and assessing the information you think they need to know. It can also help you realise how each participant in a course can be a teacher as well as a learner. Your job as a 'teacher' can change from being 'the source of knowledge' to being an influencer and role model of class culture, connecting with students in a personal way that addresses their own learning needs, and moderating discussions and activities in a way that collectively leads students towards the learning goals of the class.

Obviously Moodle doesn't force this style of behaviour, but this is what it is best at supporting. In future, as the technical infrastructure of Moodle stabilises, further improvements in pedagogical support will be a major direction for Moodle development.

4   Installing Moodle

This guide explains how to install Moodle for the first time. It goes into some detail about some of the steps, in order to cover the wide variety of small differences between web server setups, so this document may look long and complicated. Don't be put off by this - I usually set Moodle up in a few minutes!

4.1   Requirements

Moodle is primarily developed in Linux using Apache, MySQL and PHP (also sometimes known as the LAMP platform), but is also regularly tested with PostgreSQL and on Windows XP, Mac OS X and Netware 6 operating systems

The requirements for Moodle are as follows:

☐ Web server software. Most people use Apache, but Moodle should work fine under any web server that supports PHP, such as IIS on Windows platforms.

☐ PHP scripting language (version 4.1.0 or later), with the following settings:

▷ GD library turned ON, with support for JPG and PNG formats

▷ zlib library turned ON (if you want to use backup/restore on Windows)

▷ Sessions support turned ON

▷ File uploading turned ON

▷ Safe Mode must be turned OFF (see the forums on moodle.org for problems caused by Safe Mode)

☐ a working database server: MySQL or PostgreSQL are completely supported and recommended for use with Moodle 1.1. All other databases will be supported fully in the next release.

Most web hosts support all of this by default. If you are signed up with one of the few webhosts that does not support these features ask them why, and consider taking your business elsewhere.

If you want to run Moodle on your own computer and all this looks a bit daunting, then please see our guide: Installing Apache, MySQL and PHP. It provides some step-by-step instructions to install all this on most popular platforms.

4.2   Download

There are two ways to get Moodle, as a compressed package and via CVS. These are explained in detail on the download page: http://moodle.org/download/

After downloading and unpacking the archive, or checking out the files via CVS, you will be left with a directory called "moodle", containing a number of files and folders.

You can either place the whole folder in your web server documents directory, in which case the site will be located at http://yourwebserver.com/moodle, or you can copy all the contents straight into the main web server documents directory, in which case the site will be simply http://yourwebserver.com.

4.3   Site structure

Here is a quick summary of the contents of the Moodle folder, to help get you oriented:

config.php - the ONLY file you need to edit to get started

version.php - defines the current version of Moodle code

index.php - the front page of the site

- ☐ admin/ - code to administrate the whole server
- ☐ auth/ - plugin modules to authenticate users
- ☐ course/ - code to display and manage courses
- ☐ doc/ - help documentation for Moodle (eg this page)
- ☐ files/ - code to display and manage uploaded files
- ☐ lang/ - texts in different languages, one directory per language
- ☐ lib/ - libraries of core Moodle code
- ☐ login/ - code to handle login and account creation
- ☐ mod/ - all Moodle course modules
- ☐ pix/ - generic site graphics
- ☐ theme/ - theme packs/skins to change the look of the site.
- ☐ user/ - code to display and manage users

4.4   Create a data directory

Moodle will also need some space on your hard disk to store uploaded files, such as course documents and user pictures.

Create a directory for this purpose somewhere. For security, it's best that this directory is NOT accessible directly via the web. The easiest way to do this is to simply locate it OUTSIDE the web directory, otherwise protect it by creating a file in the data directory called .htaccess, containing this line:

```
{\sf deny from all}
```

To make sure that Moodle can save uploaded files in this directory, check that the web server software (eg Apache) has permission to write to this directory. On Unix machines, this means setting the owner of the directory to be something like "nobody" or "apache".

On many shared hosting servers, you will probably need to restrict all file access to your "group" (to prevent other webhost customers from looking at or changing your files), but provide full read/write access to everyone else (which will allow the web server to access your files). Speak to your server administrator if you are having trouble setting this up securely.

4.5   Create a database

You need to create an empty database (eg "moodle") in your database system along with a special user (eg "moodleuser") that has access to that database (and that database only). You could use the "root" user if you wanted to, but this is not recommended for a production system: if hackers manage to discover the password then your whole database system would be at risk, rather than just one database.

Example command lines for MySQL:

```
# mysql -u root -p
   > CREATE DATABASE moodle;
   > GRANT SELECT,INSERT,UPDATE,DELETE,CREATE,DROP,INDEX,ALTER ON moodle.*
          TO moodleuser@localhost IDENTIFIED BY 'yourpassword';
   > quit
   # mysqladmin -p reload
```

Example command lines for PostgreSQL:

```
# su - postgres
   > psql -c "create user moodleuser createdb;" template1
   > psql -c "create database moodle;" -U moodleuser template1
   > psql -c "alter user moodleuser nocreatedb;" template1
```

(For MySQL I highly recommend the use of phpMyAdmin to manage your databases - you can do all this via a web interface).

As of version 1.0.8, Moodle now supports table prefixes, and so can safely share a database with tables from other applications.

4.6   Check your web server settings

Firstly, make sure that your web server is set up to use index.php as a default page (perhaps in addition to index.html, default.htm and so on).

In Apache, this is done using a DirectoryIndex parameter in your httpd.conf file. Mine usually looks like this:

```
DirectoryIndex index.php index.html index.htm
```

Just make sure index.php is in the list (and preferably towards the start of the list, for efficiency).

Secondly, **if you are using Apache 2**, then you should turn on theAcceptPathInfo variable, which allows scripts to be passed arguments like http://server/file.php/arg1/arg2. This is essential to allow relative links between your resources, and also provides a performance boost for people using your Moodle web site. You can turn this on by adding these lines to your "httpd.conf" file.

```
AcceptPathInfo on
```

Thirdly, Moodle requires a number of PHP settings to be active for it to work. **On most servers these will already be the default settings**. However, some PHP servers (and some of the more recent PHP versions) may have things set differently. These are defined in PHP's configuration file (usually called "php.ini"):

```
magic_quotes_gpc = 1      (preferred but not necessary)
magic_quotes_runtime = 0     (necessary)
file_uploads = 1
session.auto_start = 0
session.bug_compat_warn = 0
```

If you don't have access to httpd.conf or php.ini on your server, or you have Moodle on a server with other applications that require different settings, then don't worry, you can still OVERRIDE all of the default settings.

To do this, you need to create a file called **.htaccess** in Moodle's main directory that contains definitions for these settings. This only works on Apache servers and only when Overrides have been allowed.

```
DirectoryIndex index.php index.html index.htm

<IfDefine APACHE2>
     AcceptPathInfo on
</IfDefine>

php_flag magic_quotes_gpc 1
php_flag magic_quotes_runtime 0
php_flag file_uploads 1
php_flag session.auto_start 0
php_flag session.bug_compat_warn 0
```

You can also do things like define the maximum size for uploaded files:

```
LimitRequestBody 0
php_value upload_max_filesize 2M
php_value post_max_size 2M
```

The easiest thing to do is just copy the sample file from lib/htaccess and edit it to suit your needs. It contains further instructions. For example, in a Unix shell:

```
cp lib/htaccess .htaccess
```

4.7   Edit config.php

Now you can edit the configuration file, config.php, using a text editor. This file is used by all other files in Moodle.

To start with, make a copy of config-dist.php and name it config.php. We do this so that your config.php won't be overwritten in case you upgrade Moodle later on.

Edit config.php to specify the database details that you just defined (including a table prefix - notice that this is REQUIRED for PostgreSQL), as well as the site address, file system directory and data directory. The config file itself has detailed directions and examples.

Once you have done this the rest of the installation is via a web interface. For the rest of this installation document we will assume your site is at:http://example.com/moodle

4.8   Go to the admin page

The admin page should now be working at: http://example.com/moodle/admin. If you try and access the front page of your site you'll be taken there automatically anyway. The first time you

access this admin page, you will be presented with a GPL "shrinkwrap" agreement with which you must agree before you can continue with the setup.

(Moodle will also try to set some cookies in your browser. If you have your browser set up to let you choose to accept cookies, then you must accept the Moodle cookies, or Moodle won't work properly.)

Now Moodle will start setting up your database and creating tables to store data. Firstly, the main database tables are created. You should see a number of SQL statements followed by status messages (in green or red) that look like this:

```
CREATE TABLE course ( id int(10) unsigned NOT NULL auto_increment, category int(10)
unsigned NOT NULL default '0', password varchar(50) NOT NULL default '', fullname
varchar(254) NOT NULL default '', shortname varchar(15) NOT NULL default '', summary
text NOT NULL, format tinyint(4) NOT NULL default '1', teacher varchar(100) NOT NULL
default 'Teacher', startdate int(10) unsigned NOT  NULL default '0', enddate int(10) unsigned
NOT NULL default '0', timemodified int(10) unsigned NOT  NULL default '0', PRIMARY KEY
(id)) TYPE=MyISAM


SUCCESS

...and so on, followed by:  Main databases set up successfully.
```

If you don't see these, then there must have been some problem with the database or the configuration settings you defined in config.php. Check that PHP isn't in a restricted "Safe Mode" (commercial web hosts sometimes have safe mode turned on). You can check PHP variables by creating a little file containing ¡? phpinfo() ?¿ and looking at it through a browser. Check all these and try this page again.

Scroll down the very bottom of the page and press the "Continue" link.

Next you will see a similar page that sets up all the tables required by each Moodle module. As before, they should all be **green**.

Scroll down the very bottom of the page and press the "Continue" link.

You should now see a form where you can define more configuration variables for your installation, such as the default language, SMTP hosts and so on. Don't worry too much about getting everything right just now - you can always come back and edit these later on using the admin interface. Scroll down to the bottom and click "Save changes".

If (and only if) you find yourself getting stuck on this page, unable to continue, then your server probably has what I call the "buggy referrer" problem. This is easy to fix: just turn off the "secureforms" setting, then try to continue again.

The next page is a form where you can define parameters for your Moodle site and the front page, such as the name, format, description and so on. Fill this out (you can always come back and

change these later) and then press "Save changes".

Finally, you will then be asked to create a top-level administration user for future access to the admin pages. Fill out the details with your own name, email etc and then click "Save changes". Not all the fields are required, but if you miss any important fields you'll be re-prompted for them.

```
Make sure you
remember the username
and password you chose
for the administration
user account, as they
will be necessary to
access the
administration page in
future.
```

(If for any reason your install is interrupted, or there is a system error of some kind that prevents you from logging in using the admin account, you can usually log in using the default username of "**admin**", with password "**admin**".)

Once successful, you will be returned to home page of your site. Note the administration links that appear down the left hand side of the page (these items also appear on a separate Admin page) - these items are only visible to you because you are logged in as the admin user. All your further administration of Moodle can now be done using this menu, such as:

- ☐ creating and deleting courses

- ☐ creating and editing user accounts

- ☐ administering teacher accounts

- ☐ changing site-wide settings like themes etc

4.9   Set up cron

Some of Moodle's modules require continual checks to perform tasks. For example, Moodle needs to check the discussion forums so it can mail out copies of posts to people who have subscribed.

The script that does all this is located in the admin directory, and is called cron.php. However, it can not run itself, so you need to set up a mechanism where this script is run regularly (eg every five or ten minutes). This provides a "heartbeat" so that the script can perform functions at periods defined by each module.

Note that the machine performing the cron **does not need to be the same machine that is running Moodle**. For example, if you have a limited web hosting service that does not have cron,

then you can might choose to run cron on another server or on your home computer. All that matters is that the cron.php file is called regularly.

The load of this script is not very high, so 5 minutes is usually reasonable, but if you're worried about it you can reduce the time period to something like 15 minutes or even 30 minutes. It's best not to make the time period too long, as delaying mail-outs can slow down activity within the course.

First, test that the script works by running it directly from your browser:

```
http://example.com/moodle/admin/cron.php
```

Now, you need to set up some of way of running the script automatically and regularly.

Running the script from a command line

You can call the page from the command line just as you did in the example above. For example, you can use a Unix utility like 'wget':

```
wget -q -O /dev/null http://example.com/moodle/admin/cron.php
```

Note in this example that the output is thrown away (to /dev/null).

The same thing using lynx:

```
lynx -dump http://example.com/moodle/admin/cron.php > /dev/null
```

Alternatively you could use a standalone version of PHP, compiled to be run on the command line. The advantage with doing this is that your web server logs aren't filled with constant requests to cron.php. The disadvantage is that you need to have access to a command-line version of php.

```
/opt/bin/php /web/moodle/admin/cron.php
```

```
(Windows) C:\apache\php\php.exe C:\apache\htdocs\moodle\admin\cron.php
```

Automatically running the script every 5 minutes

On Unix systems: Use **cron**. Edit your cron settings from the commandline using "crontab -e" and add a line like:

```
*/5 * * * * wget -q -O /dev/null http://example.com/moodle/admin/cron.php
```

Usually, the "crontab" command will put you into the 'vi' editor. You enter "insert mode" by pressing "i", then type in the line as above, then exit insert mode by pressing ESC. You save and exit by typing ":wq", or quit with saving using ":q! "(without the quotes).

On Windows systems: The simplest way is to use this little packagemoodle-cron-for-windows.zip which makes this whole thing very easy. You can also explore using the built-in Windows feature for "Scheduled Tasks".

On web hosts: Your web-based control panel may have a web page that allows you to set up this cron process. Ask your administrator for details on how it works.

4.10   Create a new course

Now that Moodle is running properly, you can create a course.

Select "Create a new course" from the Admin page (or the admin links on the home page).

Fill out the form, paying special attention to the course format. You don't have to worry about the details too much at this stage, as everything can be changed later by the teacher.

Press "Save changes", and you will be taken to a new form where you can assign teachers to the course. You can only add existing user accounts from this form - if you want to create a new teacher account then either ask the teacher to create one for themselves (see the login page), or create one for them using the "Add a new user" on the Admin page.

Once done, the course is ready to customise, and is accessible via the "Courses" link on the home page.

See the "Teacher Manual" for more details on course-building.

5   Teacher Manual

This section is a very quick guide to creating online courses with Moodle. It outlines the main functions that are available, as well as some of the main decisions you'll need to make.

5.1   Getting Started

This document assumes your site administrator has set up Moodle and given you new, blank course to start with. It also assumes you have logged in to your course using your teacher account.

Here are three general tips that will help you get started.

- ☐ **Don't be afraid to experiment:**
   feel free to poke around and change things. It's hard to break anything in a Moodle course, and even if you do it's usually easy to fix it.

- ☐ **Notice and use these little icons:**

   the edit icon lets you edit whatever it is next to.

   the help icon will provide you with a popup help window

   the open-eye icon will let you hide something from students

   the closed-eye icon will make a hidden item available

- ☐ **Use the navigation bar at the top of each page:**
   this should help remind you where you are and prevent getting lost.

5.2   Course Settings

The first thing you should do is look under the "Administration" on your course home page and click on "**Settings...**" (Note that this link, and in fact the whole Administration section is only available to you (and the site administrator). Students will not even see these links).

On the Settings page you can change a number of settings about your course, ranging from its name to what day it starts. I won't talk here about all these, as they all have a help icon next to them which explains them all in detail. However, I will talk about the most important of these - the **course format**.

The course format that you choose will decide the basic layout of your course, like a template. Moodle version 1.0 has three formats - in future there will probably be many more

Here are some screenshots of three sample courses in each of these three formats (ignore the different colours, which are set for a whole site by the site administrator):
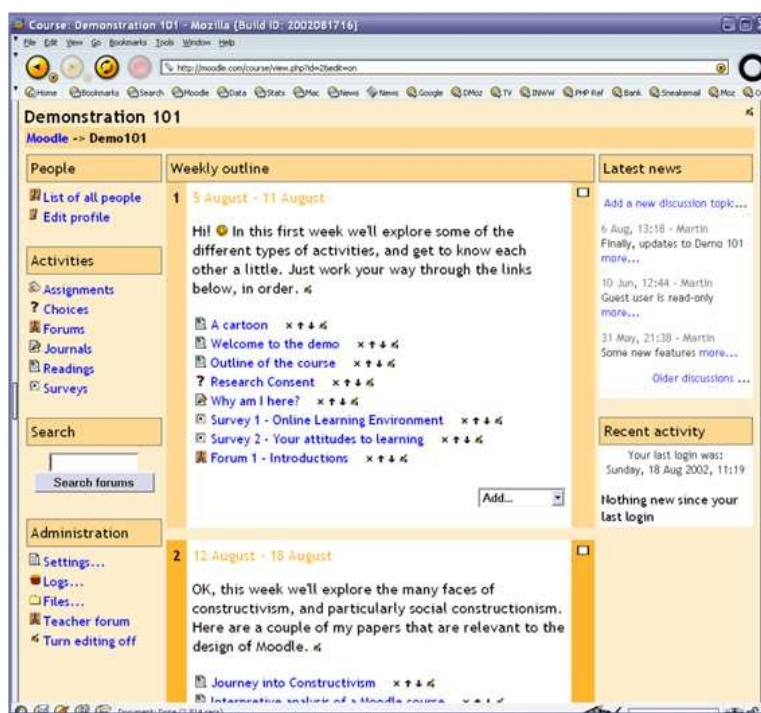
Figure 1: Weekly format

Note that the weekly and topics formats are very similar in structure. The main difference is that each box in the weekly format covers exactly one week, whereas in the topic format each box can cover whatever you like. The social format doesn't use much content at all and is based around just one forum - this is displayed on the main page.

See the help buttons on the Course Settings page for more details.

5.3   Uploading Files

You may have existing content that you want to add to your course, such as web pages, audio files, video files, word documents, or flash animations. Any type of file that exists can be uploaded into your course and stored on the server. While your files are on the server you can move, rename, edit or delete them.

All of this is achieved through the Files link in your Administration menu. The Files section looks like this: This interface is only available to teachers - it is not accessible by students. Individual files are made available to students later on (as "Resources" - see the next section).

As you can see in the screenshot, files are listed alongside subdirectories. You can create any number of subdirectories to organise your files and move your files from one to the other.

Figure 2: Topics format

Figure 3: Social format

Figure 4: Interface for uploading files.

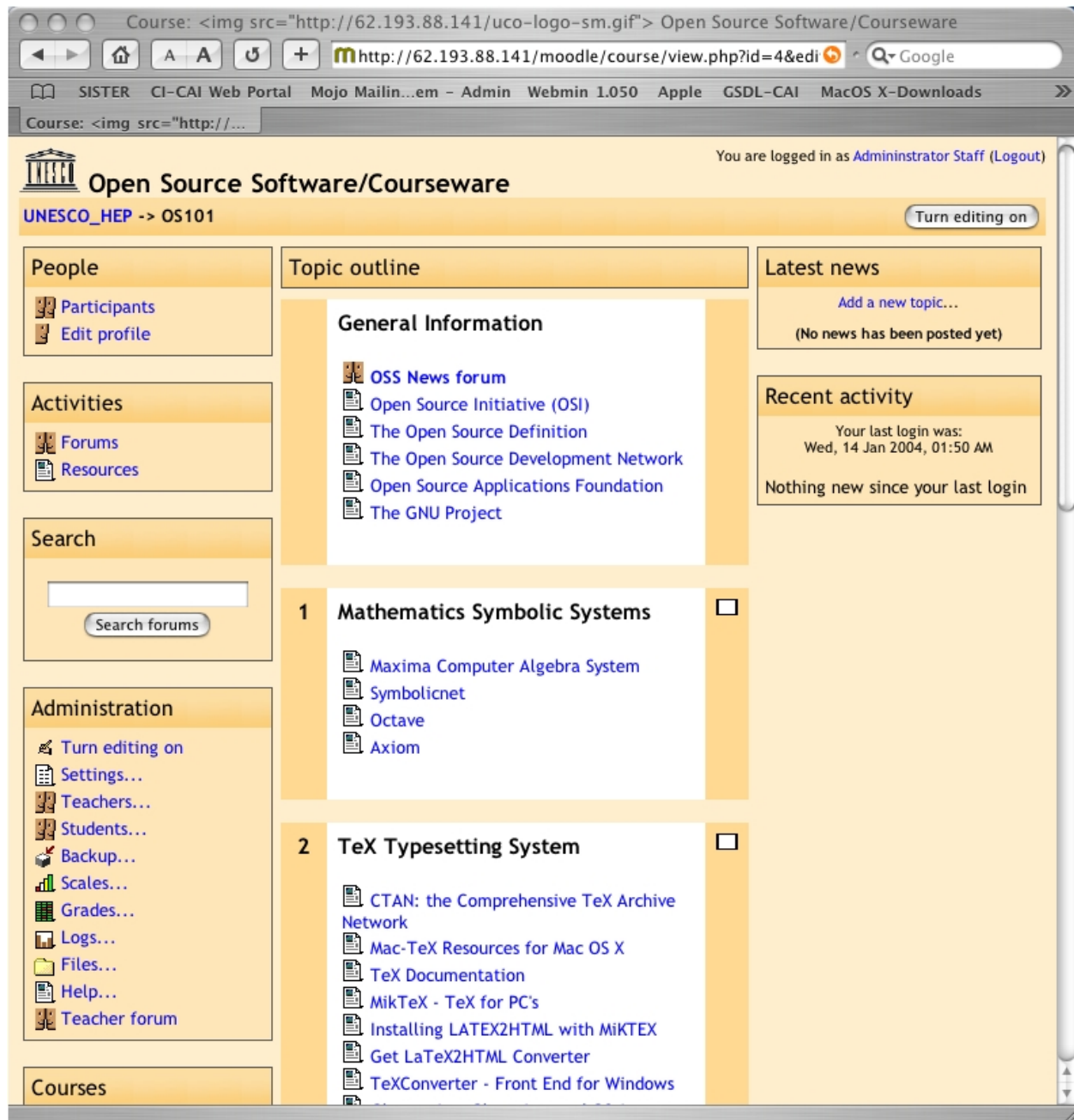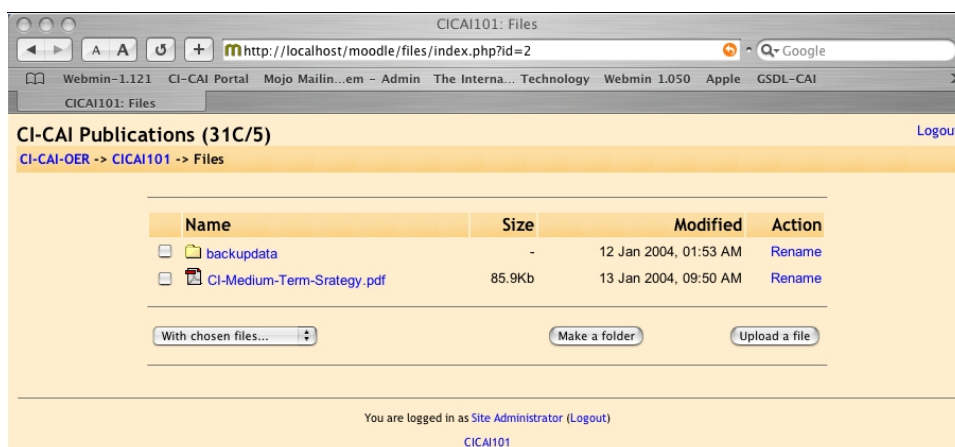Uploading files via the web is currently restricted to one file at a time. If you want to upload a lot of files at once (for example a whole web site), it can be a lot easier to use a zip program to compress them into a single file, upload the zip file and then unzip them again on the server (you will see an "unzip" link next to zip archives).

To preview any file you have uploaded just click on its name. Your web browser will take care of either displaying it or downloading it to your computer.

HTML and text files can be edited in-place online. Other files will need to be edited on your local computer and uploaded again. if you upload a file with the same name as an existing file it will automatically be overwritten.

A final note: if your content resides out on the web then you don't need to upload the files at all - you can link directly to them from inside the course (see the Resources module and the next section).

5.4   Setting up Activities

Building a course involves adding course activity modules to the main page in the order that students will be using them. You can shuffle the order any time you like.

To turn on editing, click "Turn on editing" under Administration. This toggle switch shows or hides the extra controls that allow you to manipulate your main course page. Note in the first screenshot above (of the Weekly format course) that the editing controls are turned on.

To add a new activity, simply go to the week or topic or section of the screen where you want to add it, and select the type of activity from the popup menu. Here is a summary of all the standard activities in Moodle 1.0:

Assignment An assignment is where you set a task with a due date and a maximum grade. Students will be able to upload one file to satisify the requirements. The date they upload their file is recorded. Afterwards, you will have a single page on which ou can view each file (and how late or early it is), and then record a grade and a comment. Half an hour after you grade any particular student, Moodle will automatically email that student a notification.

Choice A choice activity is very simple - you ask a question and specify a choice of responses. Students can make their choice, and you have a report screen where you can see the results. I use it to gather research consent from my students, but you could use it for quick polls or class votes.

Forum This module is by far the most important - it is here that discussion takes place. When you add a new forum, yu will presented with a choice of different types - a simple single-topic discussion, a free-for-all general forum, or a one-discussion-thread-per-user.

Journal Each journal activity is an entry in the whole course journal. For each one you can specify

an open-ended question that guides what students write, as well as a window of time in which the journal is open (weekly course format only). A general rule of thumb is to create one journal per week. Encourage students to write reflectively and critically in these journals, as they are only available to them and you. Afterwards, you will be able to grade and comment all the entries for that week or topic, and students will receive an automatic email informing them of your feedback. Journals are not designed to be continually added to - if you need to do that then add more journal activities.

Resource Resources are the content of your course. Each resource can be any file you have uploaded or can point to using a URL. You can also maintain simple text-based pages by typing them directly into a form.

Quiz This module allows you to design and set quiz tests, consisting of multiple choice, true-false, and short answer questions. These questions are kept in a categorised database, and can be re-used within courses and even between courses. Quizzes can allow multiple attempts. Each attempt is automatically marked, and the teacher can choose whether to give feedback or to show correct answers. This module includes grading facilities.

Survey The survey module provides a number of predefined survey instruments that are useful in evaluating and understanding your class. Currently they include the COLLES and the ATTLS instruments. They can be given to students early in the course as a diagnostic tool and at the end of the course as an evaluation tool (I use one every week in my courses).

After adding your activities you can move them up and down in your course layout by clicking on the little arrow icons next to each one. You can also delete them using the cross icon, and re-edit them using the edit icon.

5.5   Running the Course

There are some big plans to extend this document into a more comprehensive tutorial. Until then here are a few ideas:

1. Subscribe yourself to all the forums so you keep in touch with your class activity.

2. Encourage all the students fill out their user profile (including photos) and read them all - this will help provide some context to their later writings and help you to respond in ways that are tailored to their own needs.

3. Keep notes to yourself in the private "Teacher's Forum" (under Administration). This is especially useful when team teaching.

4. Use the "Logs" link (under Administration) to get access to complete, raw logs. In there you'll see a link to a popup window that updates every sixty seconds and shows the last hour of activity. This is useful to keep open on your desktop all day so you can feel in touch with what's going on in the course.

5. Use the "Activity Reports" (next to each name in the list of all people, or from any user profile page). These provide a great way to see what any particular person has been up to in the course.

6. Respond quickly to students. Don't leave it for later - do it right away. Not only is it easy to become overwhelmed with the volume that can be generated, but it's a crucial part of building and maintaining a community feel in your course.

6 Developer Manual

This section describes some of Moodle's design and how you can contribute. It's a bit thin at the moment - better documentation will come eventually!

6.1 Moodle architecture

From a system administrator's perspective, Moodle has been designed according to the following criteria:

1. Moodle should run on the widest variety of platforms
   The web application platform that runs on most platforms is PHP combined with MySQL, and this is the environment that Moodle has been developed in (on Linux, Windows, and Mac OS X). Moodle also uses the ADOdb library for database abstraction, which means Moodle can use more than ten different brands of database (unfortunately, though, it can not yet set up tables in all these databases - more on this later).

2. **Moodle should be easy to install, learn and modify**
   Early prototypes of Moodle (1999) were built using Zope - an advanced object-oriented web application server. Unfortunately I found that although the technology was pretty cool, it had a very steep learning curve and was not very flexible in terms of system administration. The PHP scripting language, on the other hand, is very easy to get into (especially if you've done any programming using any other scripting language). Early on I made the decision to avoid using a class-oriented design - again, to keep it simple to understand for novices. Code reuse is instead achieved by libraries of clearly-named functions and consistent layout of script files. PHP is also easy to install (binaries are available for every platform) and is widely available to the point that most web hosting services provide it as standard.

3. **It should be easy to upgrade from one version to the next**
   Moodle knows what version it is (as well as the versions of all plug-in modules) and a mechanism has been built-in so that Moodle can properly upgrade itself to new versions (for example it can rename database tables or add new fields). If using CVS in Unix for example, one can just do a "cvs update -d" and then visit the site home page to complete an upgrade.

4. **It should be modular to allow for growth**
   Moodle has a number of features that are modular, including themes, activities, interface languages, database schemas and course formats. This allows anyone to add features to the main codebase or to even distribute them separately. More on this below in the next section.

5. **It should be able to be used in conjunction with other systems**
   One thing Moodle does is keep all files for one course within a single, normal directory on the server. This would allow a system administrator to provide seamless forms of file-level access

for each teacher, such as Appletalk, SMB, NFS, FTP, WebDAV and so on. The authentication modules allow Moodle to use LDAP, IMAP, POP3, NNTP and other databases as sources for user information. Otherwise, there is work yet to do. Features planned for Moodle in future versions include: import and export of Moodle data using XML-based formats (including IMS and SCORM); and increased use of style sheets for interface formatting (so that it can be integrated visually into other web sites).

6.2 How you can contribute

As mentioned above, Moodle has a number of features that are modular. Even if you are not a programmer there are things you can change or help with.

6.2.1 Learning Activities

These are by far the most important modules, and reside in the 'mod' directory. There are seven default modules: assignment, choice, forum, journal, quiz, resource, and survey. Each module is in a separate subdirectory and consists of the following mandatory elements (plus extra scripts unique to each module):

☐ **mod.html**: a form to set up or update an instance of this module

☐ **version.php**: defines some meta-info and provides upgrading code

☐ **icon.gif**: a 16x16 icon for the module

☐ **db/**: SQL dumps of all the required db tables and data (for each database type)

☐ **index.php**: a page to list all instances in a course

☐ **view.php**: a page to view a particular instance

☐ **lib.php**: any/all functions defined by the module should be in here. If the modulename if called widget, then the required functions include:

  ▷ widget_add_instance() - code to add a new instance of widget
  ▷ widget_update_instance() - code to update an existing instance
  ▷ widget_delete_instance() - code to delete an instance
  ▷ widget_user_outline() - given an instance, return a summary of a user's contribution
  ▷ widget_user_complete() - given an instance, print details of a user's contribution
  ▷ To avoid possible conflict, any module functions should be named starting with widget_ and any constants you define should start with WIDGET_

☐ Lastly, each module will have some language files that contain strings for that module. See below.

The easiest way to start a new learning activity module is to use the template in mod/ewmodule_template.zip. Unzip it and follow the README inside.

You might also like to post first in the Activities modules forum on Using Moodle.

6.2.2   Themes

Themes (or skins) define the look of a site. A number of simple themes are provided in the main distribution, but you may want to create your own theme with your own colours, logo, styles and graphics.

Each theme is in a subdirectory of the "theme" directory, and contains at least the following files:

☐ **config.php**: defines the theme colours used throughout the site

☐ **styles.php**: the style sheet, containing CSS definitions for standard HTML elements as well as many Moodle elements.

☐ **header.html**: Included at the top of each page. This is what you need to edit to add a logo at the top of pages, for example.

☐ **footer.html**: Included at the bottom of each page.

To create your own themes for current versions of Moodle:

1. Copy one of the existing theme folders to one with a new name. I recommend starting with one of the standard themes.

2. Edit config.php and insert your own colours.

3. Edit styles.php and change your CSS styles.

4. Edit header.html and footer.html to add new logos, or change the layout.

Note that all these steps are optional - you can make a radically different look to your site simply by editing the colours in config.php

Note also that Moodle upgrades may break themes slightly, so check the release notes carefully if you are using a custom theme.

In particular, Moodle 2.0 will have a completely new display system, probably based on XSL transformations of XML output from Moodle. It is likely that the themes for this will be a completely different format, but the advantage will be a much higher possible degree of customisation (including moving elements around the page).

More discussion about this in the Themes forum on Using Moodle. If you create a nice theme that you think others might want to use, please post your zip file on the themes forum!

6.2.3   Languages

Moodle has been designed for internationalisation. Each 'string' or 'page' of text that is displayed as part of the interface is drawn from a set of language files. Each language is a subdirectory of the directory 'lang'. The structure of the lang directory is as follows:

lang/en - directory containing all files for one language (eg English)

- ☐ moodle.php - strings for main interface
- ☐ assignment.php - strings for assignment module
- ☐ choice.php - strings for choice module
- ☐ forum.php - strings for forum module
- ☐ journal.php - strings for journal module
- ☐ quiz.php - strings for quiz module
- ☐ resource.php - strings for resource module
- ☐ survey.php - strings for survey module
- ☐ .... plus other modules if any.

A string is called from these files using the get_string() or print_string() functions. Each string supports variable substitution, to support variable ordering in different languages.

eg $strdueby = get_string("assignmentdueby", "assignment", userdate($date));

If a string doesn't exist in a particular language, then the equivalent in English will automatically be used instead.

lang/en/help - contains whole help pages (for popup context-sensitive help)

- ☐ Main help pages are situated here, while help pages specific to each module are located in subdirectories with the module's name.
- ☐ You can insert a helpbutton in a page with the helpbutton function.
- ☐ eg helpbutton("text", "Click here for help about text");
- ☐ and for modules:
- ☐ helpbutton("forumtypes", "Forum types", "forum");

Note that you can edit languages online, using the administration web tools under "Check this language". This makes it easy to not to only create new languages but to refine existing ones. If you are starting a new language, please contact me, Martin Dougiamas.

You might also like to post in the Languages forum on Using Moodle.

If you are maintaining a language an ongoing basis, I can give you CVS write access to the Moodle source code so that you can directly maintain the files.

6.2.4   Database Schemas

Given a working database with defined tables, the intentionally simple SQL used in Moodle should work fine with a wide variety of database brands.

A problem exists with automatically creating new tables in a database, which is what Moodle tries to do upon initial installation. Because every database is very different, there doesn't yet exist any way to do this in a platform-independent way. To support this automation in each database, schemas can be created that list the required SQL to create Moodle tables in a particular database. These are files in lib/db and inside the dbsubdirectory of each module.

Currently, only MySQL and PostgreSQL are fully supported in this way (no-one wrote the schemas for other brands).

Moodle 1.2 will use a new method of database-independent XML schemas that will make all this unnecessary.

6.2.5   Course Formats

Moodle currently supports three different course formats: weekly, topics and social.

These are a little more connected to the rest of the code (and hence, less "pluggable") but it is still quite easy to add new ones.

If you have any ideas for different formats that you need or would like to see, get in touch with me and I'll do my absolute best to have them available in future releases.

6.2.6   Documentation and articles

If you feel like writing a tutorial, an article, an academic paper or anything else about Moodle, please do!

Put it on the web and make sure you include links to http://moodle.org/.

6.2.7   Participating in the bug tracker

Finally, I would like to invite you to register on the "bug tracker" at http://moodle.org/bugs so you can file any bugs that you find and perhaps participate in discussing and fixing them.

"Bugs" not only includes software bugs with current versions of Moodle, but also new ideas, feature requests and even constructive criticism of existing features. The beauty of open source is that anyone can participate in some way and help to create a better product for all of us to enjoy. In this project, your input is very welcome!

7   Appendix 1 - Open Source Definition

7.1   Introduction

Open source doesn't just mean access to the source code. The distribution terms of open-source software must comply with the following criteria:

1. Free Redistribution

   The license shall not restrict any party from selling or giving away the software as a component of an aggregate software distribution containing programs from several different sources. The license shall not require a royalty or other fee for such sale.

   Rationale: By constraining the license to require free redistribution, we eliminate the temptation to throw away many long-term gains in order to make a few short-term sales dollars. If we didn't do this, there would be lots of pressure for cooperators to defect.

2. Source Code

   The program must include source code, and must allow distribution in source code as well as compiled form. Where some form of a product is not distributed with source code, there must be a well-publicized means of obtaining the source code for no more than a reasonable reproduction costpreferably, downloading via the Internet without charge. The source code must be the preferred form in which a programmer would modify the program. Deliberately obfuscated source code is not allowed. Intermediate forms such as the output of a preprocessor or translator are not allowed.

   Rationale: We require access to un-obfuscated source code because you can't evolve programs without modifying them. Since our purpose is to make evolution easy, we require that modification be made easy.

3. Derived Works

   The license must allow modifications and derived works, and must allow them to be distributed under the same terms as the license of the original software.

   Rationale: The mere ability to read source isn't enough to support independent peer review and rapid evolutionary selection. For rapid evolution to happen, people need to be able to experiment with and redistribute modifications.

4. Integrity of The Author's Source Code

   The license may restrict source-code from being distributed in modified form only if the license allows the distribution of "patch files" with the source code for the purpose of modifying the program at build time. The license must explicitly permit distribution of software built from modified source code. The license may require derived works to carry a different name or version number from the original software.

   Rationale: Encouraging lots of improvement is a good thing, but users have a right to know

who is responsible for the software they are using. Authors and maintainers have reciprocal right to know what they're being asked to support and protect their reputations.

Accordingly, an open-source license must guarantee that source be readily available, but may require that it be distributed as pristine base sources plus patches. In this way, "unofficial" changes can be made available but readily distinguished from the base source.

5. <u>No Discrimination Against Persons or Groups</u>
   The license must not discriminate against any person or group of persons.

   Rationale: In order to get the maximum benefit from the process, the maximum diversity of persons and groups should be equally eligible to contribute to open sources. Therefore we forbid any open-source license from locking anybody out of the process.

   Some countries, including the United States, have export restrictions for certain types of software. An OSD-conformant license may warn licensees of applicable restrictions and remind them that they are obliged to obey the law; however, it may not incorporate such restrictions itself.

6. <u>No Discrimination Against Fields of Endeavor</u>
   The license must not restrict anyone from making use of the program in a specific field of endeavor. For example, it may not restrict the program from being used in a business, or from being used for genetic research.

   Rationale: The major intention of this clause is to prohibit license traps that prevent open source from being used commercially. We want commercial users to join our community, not feel excluded from it.

7. <u>Distribution of License</u>
   The rights attached to the program must apply to all to whom the program is redistributed without the need for execution of an additional license by those parties.

   Rationale: This clause is intended to forbid closing up software by indirect means such as requiring a non-disclosure agreement.

8. <u>License Must Not Be Specific to a Product</u>
   The rights attached to the program must not depend on the program's being part of a particular software distribution. If the program is extracted from that distribution and used or distributed within the terms of the program's license, all parties to whom the program is redistributed should have the same rights as those that are granted in conjunction with the original software distribution.

   Rationale: This clause forecloses yet another class of license traps.

9. <u>License Must Not Restrict Other Software</u>
   The license must not place restrictions on other software that is distributed along with the licensed software. For example, the license must not insist that all other programs distributed on the same medium must be open-source software.

Rationale: Distributors of open-source software have the right to make their own choices about their own software.

Yes, the GPL is conformant with this requirement. Software linked with GPLed libraries only inherits the GPL if it forms a single work, not any software with which they are merely distributed.

10. <u>License Must Be Technology-Neutral</u>
    No provision of the license may be predicated on any individual technology or style of interface.

    Rationale: This provision is aimed specifically at licenses which require an explicit gesture of assent in order to establish a contract between licensor and licensee. Provisions mandating so-called "click-wrap" may conflict with important methods of software distribution such as FTP download, CD-ROM anthologies, and web mirroring; such provisions may also hinder code re-use. Conformant licenses must allow for the possibility that (a) redistribution of the software will take place over non-Web channels that do not support click-wrapping of the download, and that (b) the covered code (or re-used portions of covered code) may run in a non-GUI environment that cannot support popup dialogues.

Origins: Bruce Perens wrote the first draft of this document as "The Debian Free Software Guidelines", and refined it using the comments of the Debian developers in a month-long e-mail conference in June, 1997. He removed the Debian-specific references from the document to create the "Open Source Definition".

Copyright 2004 by the Open Source Initiative Send questions or suggestions about the website to Steve M.:webmaster at opensource.org

8   Appendix 2 - Copyright License for Moodle

Moodle is Copyright 2001-2003, Martin Dougiamas.

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details (attached below).

8.1   GNU GENERAL PUBLIC LICENSE

[Version 2, June 1991]

Copyright (C) 1989, 1991 Free Software Foundation, Inc.

59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software–to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get

the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

<div align="center">
GNU GENERAL PUBLIC LICENSE

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION
</div>

☐ This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

☐ You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

☐ You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

▷ You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

▷ You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

▷ If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

☐ You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

▷ Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

▷ Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

    ▷ Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

☐ You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

☐ You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

☐ Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

☐ If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly

through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

☐ If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

☐ The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

☐ If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

☐ BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE

LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLD-
ERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WAR-
RANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT
LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS
FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PER-
FORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE
DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR
OR CORRECTION.

☐ IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRIT-
ING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY
AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO
YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CON-
SEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE
PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING
RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES
OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS),
EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSI-
BILITY OF SUCH DAMAGES.

<div align="center">END OF TERMS AND CONDITIONS</div>