



Digital Business Ecosystem

Contract n° 507953

Workpackage 28

DBE Training

Deliverable 28.9 -Interim

**Regional knowledge platform
implementation (Technical) blue-print and
development roadmap.**



Project funded by the European
Community under the "Information Society
Technology" Programme

Contract Number: 507953
Project Acronym: DBE
Title: Digital Business Ecosystem

Deliverable N°: 28.9
Due date: 30/04/2005
Delivery Date : 10/06/2005

Short Description: "Technical Blue-print for Knowledge Platform."

Partners owning: Intel
Partners contributed: TCH, LSE, UCE, INTEL, T6, IBM and ITA
Made available to: All project partners and EC

Versioning		
Version	Date	Author, Organisation
1.0 interim	30/05/2005	Bertrand Dory, Intel
1.1 interim	31/5/2005	Joe Butler, Intel – added executive summary

Quality check (to be carried out in)

1st Internal Reviewer:

2nd Internal Reviewer:

TABLE OF CONTENTS

Executive Summary	4
1. Introduction.....	6
2. Background	6
3. Knowledge Platform Building Block : Moodle	7
3.1. Introduction.....	7
3.2. Background.....	7
3.3. Future roadmap	7
3.4. Architecture.....	8
4. Installing Moodle	11
4.1. Requirements.....	11
4.2. Transferring files and site structure.....	11
4.3. Running the installer script	12
4.7. Continuing configuration	15
4.8. Setting up cron	18
5. Administration	20
5.1. Configuration	20
5.4. Users	22
5.5. Courses and course categories	24
5.6. Course settings	24
5.7. Logs.....	25
6. User roles	26
6.1. Role description.....	26
7. Conclusion	27

Executive Summary

The aim of this document is to provide a reference technical configuration, to support the provision and delivery of learning and knowledge content for the DBE Knowledge Platform. At the time of writing (May 2005), the initial knowledge platform has just been established, and induction material is almost ready for a pilot across 3 regions. It is expected that the pilot will provide some insights and directions for future implementations of the Knowledge Platform both at a technical level (configuration, performance) and at an operational level (communities, roles, processes). As such this document is being delivered as an interim version with scope restricted to the technical details of the initial implementation. Following an evaluation of the pilot, this document will be reissued with extended scope and revised details, before end of September 2005.

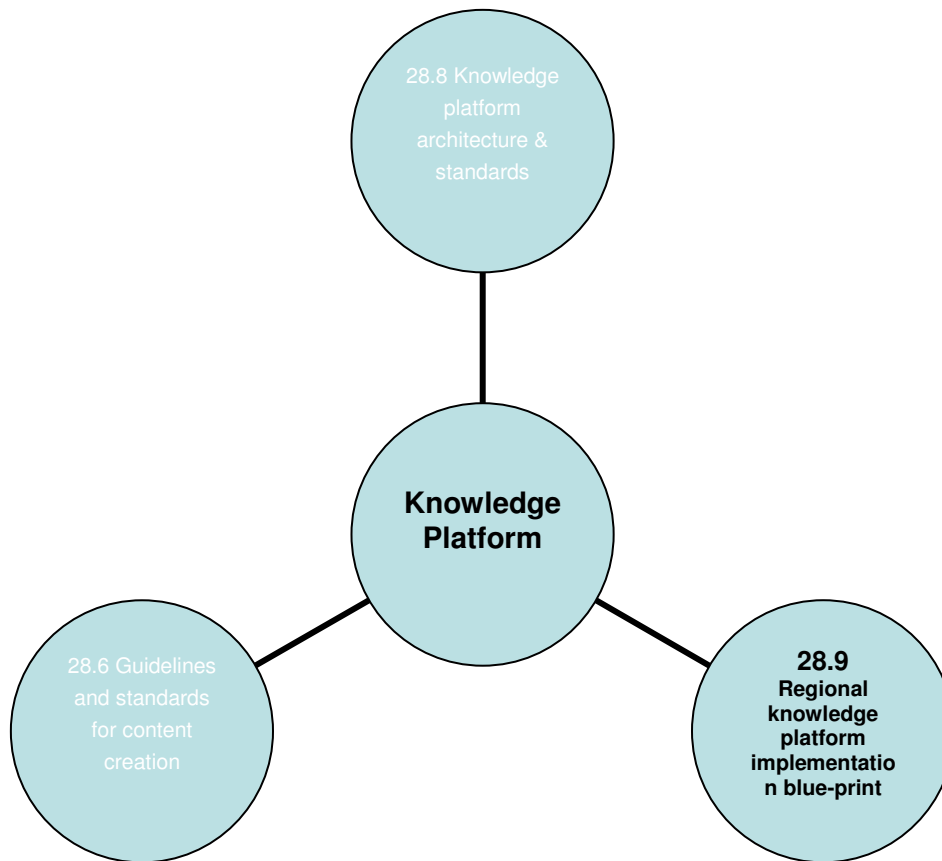
This document includes contributions from Mediamasteri – a Finnish e-learning technology company with practical expertise in the use of Moodle. Mediamasteri are hosting the implementation which will support the 3-region pilot.

The final version of this document will outline both operational and technical details of the reference implementation, and will be informed by the results of the pilot.

This document is one of three, which together describe technical and operational details of the Knowledge Platform: an open, distributed and scalable infrastructure which allows the DBE regions to create and provide learning and knowledge content which meets their specific region, but which also maximizes the opportunity for collaboration and reuse of knowledge and content across the DBE.

The three documents are :

- 28.8 Knowledge platform architecture & standards
- 28.6 Guidelines and standards for content creation
- 28.9 Regional knowledge platform implementation blue-print (this document)



The document comprises 6 sections. Sections 1 (introduction) and 2 (background) provide the context and overview. Section 3 introduces the Moodle course management system, a key building block for the Knowledge Platform. Sections 4 and 5 describe the necessary configuration and administration details. Section 6 describes the roles necessary for implementation and usage.

1. Introduction

To facilitate the dissemination and the proliferation of the knowledge, a Knowledge Platform has been designed for the DBE. This Knowledge platform is in line with the strategy defined in the D28.1 and is compliant with the guidelines defined in D28.6 and D28.8.

To ensure a rapid and wide adoption, a community based learning approach is envisaged. Communities are the key cornerstones and can be formed based on geography (regions), sectors or around trusted and respected members or facilitators. The knowledge platform will support these communities, and therefore enable the proliferation of both tacit and explicit Knowledge (please refer to D28.6).

Achieving this goal, necessitates and efficient means of creating additional Knowledge Platforms to support each emerging community.

This interim version of the blueprint describes a configuration which supports this level of interoperability and scalability.

This document does not cover the proliferation and the management of communities. This aspect will be informed by the 3-region pilot and will be covered in the final version of this blueprint. The community-based learning and the propagation of communities will be documented, and recommendations will be defined, to optimally leverage the explicit and tacit knowledge from the communities, and increase the overall knowledge capital of DBE users and developers.

2. Background

This document is in line with the general training strategy detailed in D28.1 and with the guidelines and standards defined in the deliverables D28.6 and D28.8.

Based in this strategy and on the guidelines, the Knowledge platform for the 3 initial regions has been implemented and is used for the creation and the sharing of knowledge around the DBE.

The platform can be accessed using the following link <http://dbe.moodle.fi> (which will be changed to <http://knowledge.digital-ecosystem.org>)^{*}.

^{*} Other names and brands may be claimed as the property of others.

3. Knowledge Platform Building Block : Moodle

3.1. *Introduction*

As a starting point, the course management system (CMS) Moodle* (moodle.org) has been chosen as a key building block of the Knowledge Platform. Moodle is a software package for producing internet-based courses and web sites. It's an ongoing development project designed to support learning, education and knowledge sharing.

The platform is provided freely as Open Source software (under the GNU Public License). Basically this means that the platform is copyrighted, but that user has additional freedoms. Users are allowed to copy, use and modify the platform, if they share the source to others, not modify or remove the original license and copyrights, and apply this same license to any derivative work.

The platform will run on any computer that can run PHP, and can support many types of database (particularly MySQL).

3.2. *Background*

A number of early prototypes were produced before Martin Dougiamas released version 1.0 on August 20, 2002. This version was targeted towards smaller, more intimate classes at University level, and was the subject of research case studies that closely analysed the nature of collaboration and reflection that occurred among these small groups of adult participants.

Since then there has been steady series of new releases adding new features, better scalability and improved performance.

The community has grown; more input is being drawn from a wider variety of people in different situations. The platform is used in universities high schools, primary schools, non-profit organisations, and private companies.

3.3. *Future roadmap*

As the knowledge platform gains in maturity, the community of developers and users increasingly influences its directions. Ideas, code, feedback and promotion are welcome, and user / organization can also pay to have certain features developed sooner.

Here is the current roadmap of the future, though this is always subject to change depending on sponsors and developers.

3.3.1. Version 1.5

* Other names and brands may be claimed as the property of others

The platform 1.5 aims to make the platform fully compliant with important web accessibility standards such as WAI (W3C), SENDA (UK) and Section 508 (US).

The display code will conform to XHTML Transitional 1.0, and the CSS has also been cleaned up and extended to give designers more complete control over the look and feel of their sites. This release will not yet use templates.

There is also a list of new features such as:

- Integrated instant messaging feature for direct communication
- New block format, allowing multiple copies of blocks and better block configuration (old third-party blocks need to be upgraded, see the docs in the blocks directory)
- Extended LDAP integration, LDAP can now control course creation, group assignments, user synchronisation etc
- Certain user fields can be locked by the admin
- Improved uploading, with support for automated Virus scanning of new documents using ClamAV*.
- New Tidy filter to convert all user-supplied texts into XHTML
- PAM authentication module
- Sessions can now be stored in the database (useful for The platform clusters)
- Site Policy Agreements for users
- Many other improvements and bug fixes.

3.3.2. Version 1.6

- Many of the main pages in this version will be customisable using HTML templates.
- Support for user blogs
- Improvements in structuring and maintaining the flow of courses

3.3.3. Version 2.0

This major release will contain some exciting developments in making the platform more network-aware, with a natural evolution of the platform's focus on collaboration.

3.4. *Architecture*

The platform promotes constructionist pedagogy (collaboration, activities, critical reflection, etc), and it is suitable for virtual teamwork as well as supplementing face-to-face learning. It has simple, lightweight, efficient, compatible and low-tech browser interface.

* Other names and brands may be claimed as the property of others

Course listing shows descriptions for every course on the server, including accessibility to guests. Courses can also be categorised and searched - one site can support thousands of courses.

The platform has a strong security: forms are all checked, data validated, cookies encrypted etc.

Most text entry areas can be edited using an embedded WYSIWYG editor.

An admin user, defined during setup, manages site. 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 49 languages. The code is clearly written PHP under a GPL license that is easy to modify to suit organizations needs.

Goals are to reduce admin involvement to a minimum, while retaining high security. It supports a range of authentication mechanisms through plug-in authentication modules, allowing easy integration with existing systems. Standard email method is that students can create their own login accounts, when email addresses are verified by confirmation. LDAP method means that account logins can be checked against an LDAP server. Admin can specify which fields to use. Additionally IMAP, POP3, NNTP methods are also possible; account logins are checked against a mail or news server. SSL, certificates and TLS are supported. External database can be integrated with any database containing at least two fields can be used as an external authentication source.

Each person needs only one account for the whole server, and 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.

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. Also 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 etc. Email addresses can be protected from display if required. Every user can specify own time zone, and every date in the platform is translated to that time zone. Every user can choose the language used for the platform interface (English, French, German, Spanish, Portuguese etc.)

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.

Recent changes to the course since the last login can be displayed on the course home page.

Most text entry areas can be edited using an embedded WYSIWYG editor.

Full user logging and tracking - activity reports for each student are available with graphs and details about each module as well as a detailed "story" of each students involvement including postings, journal entries etc. on one page. Mail integration is also possible so that copies of forum posts, teacher feedback etc. can be mailed in HTML or plain text.

4. Installing Moodle

4.1. Requirements*

Moodle is primarily developed in Linux operating system using Apache web server software, MySQL database server and PHP scripting language (also called "the LAMP platform"). Moodle is also regularly tested with PostgreSQL database and on Windows XP, Mac OS X and Netware 6 operating systems.

The specific requirements for Moodle are:

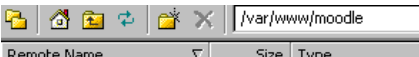
- a web server software: Apache (version 1.3 or later) or any other web server that supports PHP, such as Internet Information Server (IIS) on Windows platforms.
- PHP scripting language 4.1.0 or later (PHP 5 is supported on Moodle version 1.4. or later)
- MySQL or PostgreSQL database server (all versions). Both of these are completely supported and recommended for use with Moodle.
- MySQL or PostgreSQL client (all versions)
- PHP-MySQL module or PHP- PostgreSQL module
- PHP-GD module
- GD graphics library 1.6 or later
- zlib library
- libjpeg62 library
- libpng library

4.2. Transferring files and site structure

The basic installation version of Moodle can be downloaded as a compressed archive or via CVS. These are explained in detail on the download page: <http://moodle.org/download/>

After downloading and unpacking the archive, there will be a directory called "moodle", containing a number of files and folders. These folders and all of the contents must be placed into the web server documents directory. If the archive is transferred from local computer, it is usually better to upload it to the web server as one file, and do the unpacking there.

After unpacking the Moodle archive, the contents of the directory will be in a following way:



Remote Name	Size	Type
admin		Folder
auth		Folder
backup		Folder
blocks		Folder
calendar		Folder
course		Folder
doc		Folder
enrol		Folder
error		Folder
files		Folder
filter		Folder
grade		Folder
lang		Folder
lib		Folder
login		Folder
message		Folder
mod		Folder
pix		Folder
rss		Folder
theme		Folder
user		Folder
userpix		Folder
config.php	12,422	PHP-tiedosto
file.php	5,191	PHP-tiedosto
help.php	4,062	PHP-tiedosto
index.php	7,325	PHP-tiedosto
install.php	23,503	PHP-tiedosto
README.txt	892	Tiedosto
tags	528,177	Tiedosto
version.php	450	PHP-tiedosto

* Other names and brands may be claimed as the property of others

config.php - contains basic settings (the file will be created after a successful installation)

install.php - the script that is run to create config.php

version.php - defines the current version of Moodle code

index.php - the front page of the site

admin/ - code to administrate the whole site

auth/ - plugin modules to authenticate users

blocks/ - plugin modules for the little side blocks on many pages

calendar/ - all the code for managing and displaying calendars

course/ - code to display and manage courses

doc/ - basic help documentation for Moodle

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 the main Moodle course modules are in here

pix/ - generic site graphics

theme/ - theme packs/skins to change the look of the site

user/ - code to display and manage users

4.3. Running the installer script

To run the installer script (install.php), just try to access the Moodle URL using a web browser, or access `http://moodle_server/install.php` directly. The installer will try to set a session cookie, which must be accepted.

Moodle will detect that configuration is necessary and will lead you through some screens to help you create a new configuration file called config.php. At the end of the process Moodle will try and write the config.php file into the right location, otherwise it can be downloaded from the installer and then uploaded into the main Moodle directory on the server.

Along the way the installer will test the server environment and give suggestions about how to solve any problems. For most common issues these suggestions should be sufficient, but if the installation gets stuck, look below for more information about some of common things that might be holding you up.

4.4. General web server settings

The web server has to be set up to use index.php as a default page (perhaps in addition to index.html, default.htm and etc.).

In Apache, this is done using a DirectoryIndex parameter in your httpd.conf file. Below is an example:

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 the `AcceptPathInfo` 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.

Thirdly, Moodle requires a number of PHP settings. 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. You can often still override the default settings.

To do this, you need to create a file called `.htaccess` in Moodle's main directory that contains lines like the following. This works only on Apache servers and only when Overrides have been allowed in the main configuration.

```
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
```

4.5. Creating 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 for a test server, 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.

If you have access to Unix command lines then you can do the same sort of thing by typing commands.

Here are some example Unix 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
```

And some 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
```

4.6. Creating a data directory

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

The Moodle installer tries to create this directory for you but if it fails then you will have to create a directory for this purpose manually.

For security, it's best that this directory is NOT accessible directly via the web. The easiest way to do this is simply to locate it OUTSIDE the web directory, but if you must have it in the web directory then protect it by creating a file in the data directory called .htaccess, containing this line:

```
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 read, write and execute in this directory.

On Unix machines, this means setting the owner of the directory to be something like "nobody" or "apache", and then giving that user read, write and execute permissions.

4.7. Continuing configuration

Once the basic config.php has been correctly created in the previous step, trying to access the front page of your site will take you to the "admin" page for the rest of the configuration.

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.

Now Moodle will start setting up the 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), followed by: *Main databases set up successfully.*

```
(mysql): CREATE TABLE `mdl_course_categories` ( `id` int(10) unsigned NOT NULL auto_increment, `name` varchar(255) NOT NULL default "", `description` text NOT NULL, `parent` int(10) unsigned NOT NULL default '0', `sortorder` int(10) unsigned NOT NULL default '0', `coursecount` int(10) unsigned NOT NULL default '0', `visible` tinyint(1) NOT NULL default '1', `timemodified` int(10) unsigned NOT NULL default '0', PRIMARY KEY (`id`), UNIQUE KEY `id` (`id`)) TYPE=MyISAM COMMENT='Course categories'
```

Success

```
(mysql): CREATE TABLE `mdl_course_display` ( `id` int(10) unsigned NOT NULL auto_increment, `course` int(10) unsigned NOT NULL default '0', `userid` int(10) unsigned NOT NULL default '0', `display` int(10) NOT NULL default '0', PRIMARY KEY (`id`), UNIQUE KEY `id` (`id`), KEY `courseuserid` (course, userid)) TYPE=MyISAM COMMENT='Stores info about how to display the course'
```

Success

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". You can check PHP variables by creating a little file containing `<?php phpinfo() ?>` and looking at it through a browser. Check all these and try this page again.

Scroll down to the very bottom of the page and click 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 etc. You can always come back and edit these later on using the admin interface. Scroll down to the bottom and click "Save changes".

Variables

lang:	<input style="width: 90%;" type="text" value="English (en)"/>	<p>Choose a default language for the whole site. Users can override this setting later.</p>
langmenu:	<input style="width: 90%;" type="text" value="Yes"/>	<p>Choose whether or not you want to display the general-purpose language menu on the home page, login page etc. This does not affect the user's ability to set the preferred language in their own profile.</p>
langlist:	<input style="width: 90%;" type="text"/>	<p>Leave this blank to allow users to choose from any language you have in this installation of Moodle. However, you can shorten the language menu by entering a comma-separated list of language codes that you want. For example: en,es,fr,it</p>
locale:	<input style="width: 90%;" type="text" value="en"/>	<p>Choose a sitewide locale - this will affect the format and language of dates. You need to have this locale data installed on your operating system. (eg en_US or es_ES). If you don't know what to choose leave it blank.</p>
timezone:	<input style="width: 90%;" type="text" value="Server's local time"/>	<p>You can set the default timezone here. This is the only the DEFAULT timezone for displaying dates - each user can override this by setting their own in their profile. "Server time" here will make Moodle default to the server's operating system setting, but "Server time" in the user profile will make the user default to this timezone setting.</p>
country:	<input style="width: 90%;" type="text" value="United Kingdom"/>	<p>If you set a country here, then this country will be selected by default on new user accounts. To force users to choose a</p>

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

Next you will see more pages that print lots of status messages as they set up all the tables required by the various Moodle module. As before, they should all be green.

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

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 etc. Fill this out (you can always come back and change these later) and then press "Save changes".

Finally, you will 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.

User profile for Admin User

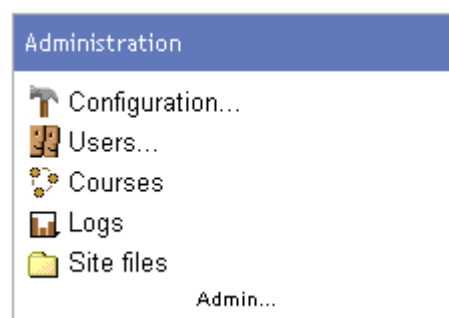
Username:	<input type="text" value="admin"/>
New password:	<input type="password"/> (Leave blank to keep current password)

Given name:	<input type="text" value="Admin"/>
Surname:	<input type="text" value="User"/>
Email address:	<input type="text" value="admin@moodle.fi"/>
Email display:	<input type="text" value="Allow everyone to see my email address"/>
Email activated:	<input type="text" value="This email address is enabled"/>
Email format:	<input type="text" value="Pretty HTML format"/>
Email digest type:	<input type="text" value="No digest (single email per forum post)"/>

Make sure that you remember the username and password you chose for the administration user account, as they will be necessary when accessing the administration page in future.

(If your install is interrupted for some reason, 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 the installation is done successfully, you will be returned to the home page of your new 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 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.

4.8. Setting 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 cannot 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. This kind of regular mechanism is known as a cron service.

Note that the machine performing the cron does not need to be the same machine that is running Moodle. 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://moodle_server/moodle/admin/cron.php`

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

On Windows systems

The simplest way is to use this little package `moodle-cron-for-windows.zip` that makes this whole thing very easy by installing a small Windows service.

Using the command line on Unix

There are different command line programs you can use to call the page from the command line. Not all of them may be available on a given server.

For example, you can use a Unix utility like 'wget':

`wget -q -O /dev/null http://moodle_server/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://moodle_server/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
```

Using the crontab program on Unix

If you have a command line, you can set up crontab using the command:

```
crontab -e
```

and then adding one of the above commands like:

```
*/* * * * * wget -q -O /dev/null http://moodle_server/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 without saving using ":q!" (without the quotes).

5. Administration

5.1. Configuration

5.1.1. Variables

The Administrator configures variables that affect general operation of the knowledge platform.

The most important variables are explained below.

- **Language settings:** A default language must be defined for the whole platform. Users can override this in their own settings. The general-purpose language menu can be either displayed or hidden. Defining a list of available languages may also shorten the.
- **Time zone:** the default time zone for the whole platform. Users can override this in their own settings.
- **Country:** a country that will be used as default for new user accounts
- **SMTP host(s):** the full name of one or more SMTP servers that Moodle will use to send mail (from discussion forums etc.)
- **HTML editor:** An embedded HTML text editor can be allowed or not allowed. Users may also choose whether to use the editor or not.
- **Maximum editing time:** the amount of time that users can re-edit their own forum postings, journal feedback etc. 30 minutes is a normal setting.
- **Assigning Creators:** If this is allowed, Creators can assign other Creators within their courses. Otherwise Administrators will control this.
- **Allow unenroll:** If this is allowed, users can unenroll themselves from courses whenever they like. If not allowed, then only Creators and Admins can unenroll users.
- **Maximum upload size:** This specifies the maximum size of uploaded files. The upload size can be further limited at course level or module level.
- **Force login:** Normally, people can read the front page of the platform with course listings without logging in first. If this setting is enabled, people are forced to log in before seeing anything on the site.
- **Enable shared folders:** This will enable file manager's shared folder feature, so that folders and files can be shared between courses.

5.1.2. Site settings

The name of the platform, front-page description, and front-page format are defined on this page. Also the default words for Course Creators and Users are defined here.

5.1.3. Themes

Moodle themes are selected on this page. Themes (or skins) define the look of the Moodle platform. A number of simple themes are provided in the main distribution version, but special themes can be designed with their own

colours, logo, styles and graphics. All of this cannot be done in Moodle. There is a "theme" directory under Moodle installation, which contains all themes and the files to be edited to create a different look for the platform. Most of this editing is made with PHP and CSS style sheets.

```
.cal_popup_fg {
    background-color: <?php echo $THEME->cellcontent?>;
}

.cal_popup_bg {
    border-top: 2px <?php echo $THEME->cellcontent2 ?> solid;
    border-left: 2px <?php echo $THEME->cellcontent2 ?> solid;
    border-right: 2px <?php echo $THEME->cellheading2 ?> solid;
    border-bottom: 2px <?php echo $THEME->cellheading2 ?> solid;
    background-image: url(<?PHP echo "$themeurl"?>/gradient.jpg);
    color: #FFFFFF;
```










5.1.4. Language

The entire language pack can be checked and edited directly from Moodle. On this page you can edit all of the available languages, and replace the default Moodle language with your own words and phrases.

5.1.5. Modules and blocks

These pages list all of the installed activity modules and side blocks. Modules/blocks can be visible, hidden or completely deleted. Also the specific settings of some modules can be defined here. Usually the deletion of a module or block is not necessary - the selection of available activity modules and side blocks on a course consists of those that are made visible by the Administrator.

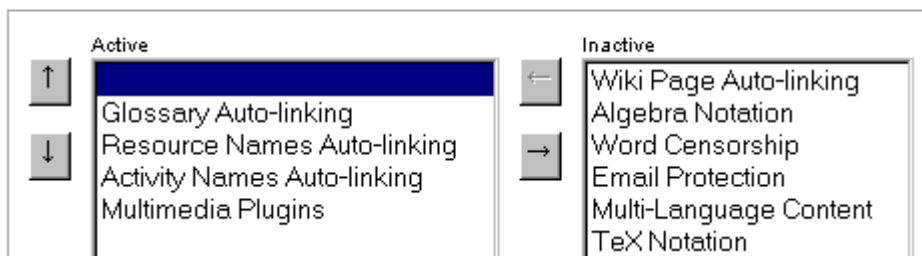
Modules

Activity module	Activities	Version	Hide/Show	Delete	Settings
 Assignment	0	2004060400		Delete	Settings
 Chat	1	2004060400		Delete	Settings
 Choice	0	2004070102		Delete	
 Dialogue	0	2004013101		Delete	
 Forum	23	2004081800			Settings
 Glossary	0	2004080900		Delete	Settings

Filters

Moodle filters are text filters that help Moodle analyze the text in a course. There are a number of filters that have their specific functions, such as automatic linking of resource/activity names or glossary words in any text within a Moodle course.

Filters

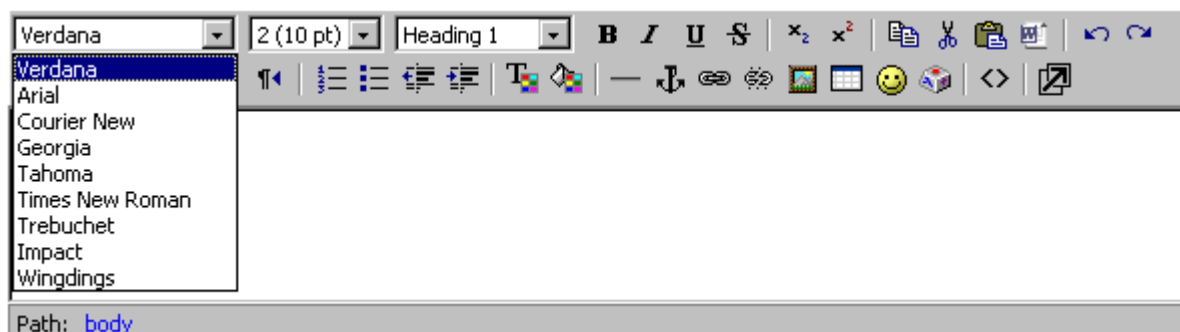


5.2. Backup

Moodle has an automated backup system that can be scheduled to run on a nightly basis to export all of the course materials for the entire platform. The backup tool in Moodle actually runs the same functions as an individual course backup. It simply runs on all of the courses on the platform automatically at a time designated by the Admin. It's a good idea to schedule backups for a time when the server isn't usually busy.

5.3. Editor settings

This area gives the ability to enable and customize the HTML editor. The editor default font family and font size can be configured here. Also the available fonts on the editor's dropdown menu are listed and can be changed.



5.4. Users

5.4.1. User authentication

Moodle provides the following user account authentication methods:

- **Email-based**

Email authentication is the default account authentication method. With this method, users can create their own accounts. They then receive an email at the address they specified in their account profile to confirm their account.

The email-based authentication can also be configured so that the new user has to select a course to join, when creating an account. After

that, a confirmation request email is sent to the Creator of the selected course. The new user account is created after the Course Creator has accepted the user to join the course. Requests for joining other courses can be made the same way, after logging in first.

- **Manual**
This method requires the Administrator to manually create all user accounts.
- **No authentication**
Users can create accounts with any external validation.
- **Use First Class / POP / IMAP / LDAP / NNTP server**
These methods use an external server to check a users username and password. If the name and password match the data on the mail server, an account with the same username and password is created on the platform.
- **Use an external database**
This functions much like the other external server authentication methods. The difference is it uses a database of user data, like in a student information system. Once the username and password stored in the external database have been validated, you can tell Moodle to copy additional data.

5.4.2. Add and edit user accounts

User accounts can be created manually by selecting **Add new user**. The required information is *username, password, first name, surname, email address and city/town*. **Edit user accounts** allows the Admin to edit the user profile of anyone on the system. Most frequently this is used to reset passwords for users who are unable to login.

Upload users means importing new user accounts from a text file. The user data text file must follow a certain format. The first line contains the names of the column headers, like *username* and *password*. Below the first line, each user record must be on one line and each column must be separated by commas. The user text file must have the following columns: *username, password, firstname, lastname, and email*. These columns are optional:

institution, department, city, country, lang, timezone, idnumber, icq, phone1, phone2, address, url, description, mailformat, maildisplay, htmeditor, autosubscribe.

Students can also be enrolled in courses using the text file. Add in the column headings course1, course2, etc. Then in the data file put the course short name in those columns.

5.5. Courses and course categories

The System Admins and Course Admins are responsible for adding courses to the platform. There isn't a way to automatically create courses, although this will be added in future versions of Moodle. So the courses for the Course Creators have to be added manually.

Courses can be organized by course categories. It's up to the System Admin(s) to define what categories are used, but each course can only be listed in one category. Users will find it easier to find their own courses if they are organized by descriptive categories.

Adding new categories is very simple. It can be done by clicking Courses from the Administration block on the front page. At the top of the page is a text area and an "Add new category" button. Type the name of the new category in the text area and click the button.

New courses can be added on the same Courses page, by clicking Add new course.

5.6. Course settings

Edit course settings

Category: ?

Full name: ?

Short name: ?

ID number: ?

Summary:

Verdana 1 (8 pt) Heading 1
B *I* U ~~S~~
 x_2 x^2

Write a concise and interesting paragraph here that explains what this course is about

Path: [body](#) ?

Format: ?

Course start date: ?

The platform provides a number of options for the general format of a course. A Creator can choose to order the course chronologically by week, conceptually by topic, or socially with a big forum as the central organizing principle.

The same settings area where the Creator sets the course format also gives access to a number of other course options. The course can be placed into one of the existing categories. Full name, short name and summary are informational for the Users to identify which course they are working in.

Course availability to Users can also be set on this page. This way the course can be hidden/closed from Users, without affecting the Creator access.

Once the course is added, the Creator will be able to add content and also to add other Creators. Users will be able to enroll in the course, if the course settings allow users to participate.

5.7. Logs

The Administrator has access to user activity logs for the entire platform. Logs can be viewed based on course, user, date and activity.

One of the tools in the logs area is the View Live Logs. This is a continually refreshed page displaying all user activity for the past hour. It may be used for instance as a tool to gauge server load and to get a snapshot of how Users and Creators are using the system.

6. User roles

Once users have accounts on the platform, they need access to the courses they are taking. The first step is to enroll the user in a course. Users can enroll themselves in their courses. Moodle also includes some tools to automatically enroll students in the proper courses.

But users also need to have the proper permissions once they are enrolled. Each role type is granted specific permissions, which determine what each DBE member can do on the platform.

Course Creators need to have creator permissions so they can build their courses.

6.1. *Role description*

A member with an account on the platform has (at least) one of the following roles:

6.1.1. Administrator (Admin)

Only one or two persons usually have System Administrator rights on the platform. Administrators can go anywhere and do anything on the site. Admins are responsible for defining the platform configuration and settings, and for adding Course Creators and Course Administrators. Only the main Administrator is able to add other System Administrators. Admins are also responsible for adding/editing user accounts manually, if needed, or deleting existing user accounts.

6.1.2. Course Administrator

Every DBE area has own Course Administrator(s). Course Administrators can create new courses to existing course categories. They also have the right to assign Course Creators to any course on the platform.

6.1.3. Course Creator

Course Creators have rights to add/remove and change content on a course, enroll and unenroll Users to the course, grade user activities and facilitate within activities. A Course Creator selects, which side blocks are visible to Users on a course. Also the whole course or it's content can be hidden from Users by a Creator.

6.1.4. Non-editing Creator

A "non-editing" Creator is a secondary creator or tutor that is sometimes used on a course. They are only able to grade students and facilitate within activities, but not to edit or add anything.

6.1.5. User

All DBE members are Users on the platform. If no other rights are defined, then the person is a basic User. A User can only participate on those courses that he/she is enrolled in. Users have the ability to enroll into courses and use the given tools and materials that the Creator has created. Users may interact with the tools on a course but may not add or remove tools or edit settings. Also a User cannot affect the structure of a course. If an enrollment key is in place on a course then Users cannot enroll into the course without the correct key. An enrollment key only has to be entered once.

6.1.6. Visitors (Guests)

Administrators may allow Visitors/Guests to access the platform. Visitors only have read permissions on the platform, which means that they can only look at courses and not edit them in any way. If an enrollment key is in place on a course then Visitors must enter the correct key every time they visit the course.

7. Conclusion

This document summarises the technical requirements and procedures necessary to implement an instance of the knowledge platform. Following the 3-region pilot, this document will be re-issued in its final version. The final version will address the proliferation and the management of communities. Community-based learning and the propagation of communities will be documented, and recommendations will be defined, to optimally leverage the explicit and tacit knowledge from the communities, and increase the overall knowledge capital of DBE users and developers.