# Contents

1 **Introduction** 2

1.1 Features 2

1.2 Requirements 3

1.2.1 Partially Tested 3

2 **Installation and Updating** 3

2.1 Update Procedure 3

2.2 New Installation 3

3 **Configuration** 4

3.1 config.php 4

3.2 Administration Editor 5

4 **Tutorial** 5

4.1 Adding a file 5

4.2 Adding a user 6

4.3 Authorizing a file 6

4.4 Checking out a file 6

5 **Automated Installation and Upgrade Notes (For Developers)** 6

6 **Changes** 7

6.1 1.2rc1 - July 9th, 2003 7

6.2 1.1-Final - May 6th, 2003 8

6.3 1.1rc2 - March 23, 2003 8

6.4 1.1rc1 - March 10, 2003 9

6.5 1.0 - Initial Release - December 19, 2003 9

# 1 Introduction

OpenDocMan is a full featured Web-based document management system designed to conform to ISO 17025/IEC. It features automatic installation, file expiration, multiple version handling, file check-in/out, departmental access control, file moderation, fine grained user access control, email notificaiton and a great search function. Written in PHP, and utilizing MySQL for the backend, this project is useful for any company looking to keep their documentation in a centralized repository.

## 1.1 Features

1. Automatic Installation - Automatically install the application or upgrade your current version
2. Check-in and Check-out files - This provides file locking so that no two persons can update the same file at the same time
3. Search Engine - Keyword search helps locate documents quickly
4. Multiple Users, Departments, and Categories - The system allows for an unlimited number of these elements
5. File Moderation - Allows for departmental "reviewers" to authorize or decline files for publication to the repository. Can be turned off.
6. Email Notification - The system has a built in email notification system for file authorization process
7. Fine Grained Access - Ability to control read/write/admin/forbidden permissions for individual files based on users or groups or both.
8. File History - Provides a history of actions taken upon a file
9. Admin and Root users - Each installation can have any number of "admin" users, who can create users, categories, etc.. One "root" user can do all.
10. Multiple Document Versions - Instead of overwriting a document when a new version is added, the old versions are kept, so there is a traceable change path.
11. Archiving function for deleted files - Files that are deleted are now put into an "archive" state and can be left there, un-archived, or permanently deleted.
12. Bookmarkable URL's - You can bookmark a page in OpenDocMan and email it to a co-worker, etc..
13. Breakcrumb Navigation - As you navigate through the site, breadcrumb links are generated to aid in backward navigation.
14. File Expiration - Files can be set to automatically expire after a definable period. The expiration action can be one of either Remove from file list until renewed, Show in file list but non-checkoutable, Send email to reviewer only, or Do Nothing.

## 1.2 Requirements

- Apache Webserver 1.3.x (or any other webserver, that supports PHP) (http://www.apache.org/)
- MySQL Server 3.22+ (http://www.mysql.com/)
- PHP 4+ compiled with MySQL-Support (http://www.php.net/)
- Linux

### 1.2.1 Partially Tested

- Microsoft IIS Server
- MS-Windows

# 2 Installation and Updating

## 2.1 Update Procedure

To update your current version to the latest release:

1. Rename your current opendocman folder.
2. Unarchive opendocman into a new folder
3. Edit the config.php in your new folder (Look at your old config.php file for some settings that can be transferred over from the old config to the new).
4. Load the setup.php page (ex. http://www.mydomain.com/opendocman/setup.php) and click on the appropriate upgrade link. Refer to your old config.php to see what version you were using before.

## 2.2 New Installation

1. Untar files into any dir in your webserver documents dir (ex. /var/www)
2. Edit config.php
   (a) All parameters are commented and should be self explanatory. Change any that apply, especially the database parameters.
3. If you DO have database creation permissions to your MySQL database then you can use the automatic setup script (preferred method).
   (a) Load the setup.php page and click on the new install link. (ex. http://www.mydomain.com/opendocman/setup.php)
   (b) Enter the username and password of a user that has database creation permissions for the database configured in config.php
   (c) Skip step 4 and move on to step 5

4. If you DO NOT have database creation permissions, be advised that you should be carefull in doing things manually
   (a) NOTE: The entries below are just examples.
   (b) create a MySQL-database and MySQL-User for opendocman

   ```
   $> mysql -u root -p
   Welcome to the MySQL monitor. Commands end with ; or \g.
   Your MySQL connection id is 5525 to server version: 3.22.32
   Type 'help' for help.
   mysql> create database opendocman;
   mysql> grant select,insert,update,delete,create on opendocman.* to opendocman@localhost identi
   mysql> flush privileges;
   mysql> exit;
   $> mysql -u opendocman -p opendocman < database.sql
   ```

5. Make a directory for the files to be stored that is accessible to the web server but not available by browsing

   ```
   $>mkdir /usr/local/opendocman/data
   ```

6. Point your favorite webbrowser to the opendocman folder: ex. "http://www.mydomain.com/opendocman"
7. Login as "admin" (without password). After that, go to "admin->users->update->admin" and set your admin password.
8. Add departments, categories, users, etc.
9. Enjoy!

# 3 Configuration

## 3.1 config.php

Most of the configuration for OpenDocMan is controlled by a file name config.php. Below are some of the configurable options in that file:

- $database - The name of the database used (ex. opendocman)
- $user - The user that has update/insert/delete permissions on above database
- $pass - Password for above user
- $hostname - The hostname of the database server (ex. db1.mydomain.com)
- demo - This setting is for a demo installation, where random people will be all loggging in as the same username/password like 'demo/demo'.

- authen - Future setting to control other authentication methods. Currently only MySQL and web-based kerberos authentication is supported.
- page_limit - Set the number of files that show up on each page
- displayable_len - Set the maximum displayable length of text field in file list
- base_url - Set this to the url of the site (no ending slash)
- title - This is the browser window title and be anything you want (ex. Mycomany Document Managment System)
- current_version - This is the program version for window title (This should be set to the current version of the program)
- site_mail - The email address of the administrator of this site. Users will be told to contact this person if they encounter errors.
- root_username - This variable sets the root username. It must be set to one of the current users of opendocman. The root user will be able to access all files and have authority for everything. (ex. jsmith)
- revision_expiration [integer]- Period in days that files will expire.
- file_expired_action [1,2,3,4] - Action to take on files that expire.
- authorization [On, Off] - Turn file moderation on or off.
- secureurl [On,Off] - Hide the URL contents from prying eyes
- dataDir - Location of file repository. This should ideally be outside the Web server root. Make sure the server has permissions to read/write files in this folder!
- $allowedFileTypes - This is a list of file types that will be allowed to be added to the repository. This may need some tweeking depending on what types of files your users have.

## 3.2 Administration Editor

When a user is set up as an "admin", they will see an admin menu in the upper right while they are logged in. There are options for User, Category, Department, and File Management under this menu item.

# 4 Tutorial

## 4.1 Adding a file

1. Click on the "Add" icon in the menu area
2. Click on the "Browse" button and select the file to upload
3. Choose a category for the file. This will be used during file searches to help narrow the search result.

4. Setup the default departmental permissions. You may set up individual departments with different settings, set all departments to the same setting, and/or select default permissions for any unset department. The program will remember your settings as you go, so as you select each department, the current permission will show.
5. Enter a description. This will be used you can go back to set department and alter the setting and will display in the main file list.
6. Add any comments necessary for users of the file.
7. Set specific user premissions, if any. You may shift-click, ctrl-click, or click on any of the usernames in any of the columns to specify certain rights to certain users of the file.
8. Click on "Add Document"
9. If you have file moderation turned on, you will now have to authorize the file before it is available for viewing. If file moderation is turned off, you should see the file in the main file list now.

### 4.2 Adding a user

### 4.3 Authorizing a file

### 4.4 Checking out a file

]Credits

Originally inspired by an article called cracking the vault.

## 5 Automated Installation and Upgrade Notes (For Developers)

There is a new folder named "install" which contains files use by the new script called setup.php. This is an automated web-based update/installation script. Here is how it works for users:

1. The user loads setup.php into their browser. They can either select the new installation link, or one of the upgrade links.
2. For a new installation:
   (a) The user will be prompted to enter a priviledged mysql username and password. This is for the database creation and grant assignments. The script will then proceed to install all the necessary data structures and default data entries for the most current version of ODM.
3. For updates:
   (a) The user will need to find their current version (which comes from their old config.php), and they would then click on the appropriate upgrade link. For example, if their version number is 1.0, they would click on the "Upgrade from 1.0" link. This will apply all necessary database changes to their current database.

For developers, when there is a new version release, a few new files need to be created and a current files modified.

1. upgrade_x.php - where x is the release name. This file should follow the same format as the other upgrade_x.php files and is used for upgrades only. This should be built from the output of a program like mysqldiff.pl and is the "diff"erence between the it's version, and the version before it.

2. Setup.php - add a new function for the new version upgrade (ex. "do_update_x()") where x is the release name.

   (a) Inside this new function, you must "include" each previous upgrade file in succession (see upgrade_10.php for an exmaple, which is for a user upgrading from 1.0 to the latest).

   (b) Add a new case statement for the new upgrade call

   (c) Add a new link to print_intro() for the upgrade function

3. odm.php - This file should follow the same format as the current odm.php file, and should contain all the necessary sql commands to create a new installation based on the latest database schema.

4. database.sql - This should contain the same sql commands as odm.php, only in a mysqldump format for users that need to manually install the program for some reason. This can be a mysqldump straight from the latest database schema.

These files MUST be kept syncronized for each release!

# 6 Changes

## 6.1 1.2rc1 - July 9th, 2003

- Better permission: new design for this page will be both more straight forward for the user and the programmers. The permission select boxes are now used to directly specify which user has which right to the file. Admin needs not to set inheriting rights. When the admin sets rights to the department, he indirectly set inheriting rights for all users. Require more testing.

- Multiple version of document: Instead of overwriting a document when a new version is added, keep the old versions, so there is a traceable change path. Revision configuration is in config.php Revision dir will be created if not. Inside revision folder, for each file, a folder will be created with their name (e.g 80/ for 80.dat). Inside 80 folder for example, there will files with the naming of format of 80_x.dat were x is the revision number. User can view these older revision through details.php

- Add archiving function for deleted files: When file is deleted by user, they are moved into an archiving folder. This folder is configurable in config.php. Root user can permenently delete the files in admin.php under Del/Undel link

- Add ability to change owner of file: Admin can change owner in edit.php. New or old owner can have their rights revolked

- File Expiration: file expiration script's name is check_exp.php. This script will run under root authority. Anyone can invoke it if they know its url. A link to this page is in Admin.php. Configuration to this script is available in config.php under "revision_expiration"

- Enable/Disable Reviewing/Authorization: User can disable the Review and Authorization proccess in config.php setting under "authorization". After disabled, when files are added or check-ined, they by pass the whole Review and Authorization proccess

- Drop the Comment column in out.php: Comments columns are droped out of any table listing
- bookmarkable URL: Url are now bookmarkable
- Wildcard search: Wild card search are now integrated into search.php. To search for the key word "Administration", user can type in Admin.
- add bread crumb: Bread Crums are added to the project for betting browsing
- Should allow telephone numbers in international format: telephone numbers enforcement are removed
- Add a timeout function to checked-out document: In admin.php, root user can see all the checked out files and clear their check-out status
- 3-4 seconds for 5000 files with DB of 3.4 mb in size: ODM1.2-rc1 is optimized to handle a heavy amount of data in a few seconds

## 6.2 1.1-Final - May 6th, 2003

- Added icons to the review notices
- changed from using copy() to move_uploaded_file() instead
- New users guide added to docs/
- Small database schema changes

## 6.3 1.1rc2 - March 23, 2003

- This release fixes a number of bugs with user sessions, url encoding, file viewing, and others. See changelog for details.\ Also added a demo mode setting which will make it a "read only" installation for demo purposes, where the users will not be able to modify their info (password, phone, etc). Added new icons for notices.
- setup.php, commitchange.php: fixed username issues from bug #702453
- config.php: added a demo mode setting
- functions.php: fix urlencode for sort_browser. Resolving bug-702377
- view_file.php: added backslash thanks Huy
- out.php, images/exclamation.gif, images/exclamation_red.gif: added yellow ! for new notice added red ! for rejected notice
- view_file.php: fix https viewing and downloading problem
- FormCheck.js: phonenumber validation has been removed

## 6.4 1.1rc1 - March 10, 2003

- New: web based installer/updater
- New: Ability to run with register_globals=off
- New: New mysql connection method
- New: error_reporting can be set to any level
- New: short_open_tags can be off.
- edit.php: fixed error: unable to set modify permission to users
- toBePublished.php: fixed PHP_SELF
- databaseData_class.php: a new level of abstraction has been added to this foundation class. Unlike before, this class's DB information such as database name, and table names can now be configured. All derived classes inherit this feature. One big drawback is the readability every query has been dramatically reduced.
- FileData_class.php: fixed error in sql statement
- search.php: fixed some locals that were set to _GET
- add.php: turned off error rep for status_bar
- search.php: fixed undefined var $cases in 119
- view_file.php: fixed bug in split code. Missing backslashe for period
- install/: odm1.1rc1.php, upgrade.php, upgrade_10.php: initial import of installation scripts
- database.sql: fixed column types (tinyint to smallint, etc.)

## 6.5 1.0 - Initial Release - December 19, 2003