

DCMMS Development Guide

Maher Abdel Karim

Steffen Macke

Maher Abdel Karim

Abdulqader Jaradat

DCMMS Development Guide

Maher Abdel Karim
Steffen Macke
Maher Abdel Karim
Abdulqader Jaradat

Published \$Date: 2008-10-20 10:34:18 +0200 (Ø§Ù#Ø§Ø«Ù#Ù#Ù#, 20 Ø°Ø´Ø±Ù#Ù#Â Ø§Ù#Ø£Ù#Ù# 2008) \$
Copyright © 2003 - 2005 DORSCH Consult, NGWA
Copyright © 2006-2008 DORSCH Consult WATER and Environment
Copyright © 2008 Steffen Macke

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is available from the Free Software Foundation (<http://www.gnu.org>).

Table of Contents

DCMMS Development	1
1. Introduction	1
2. Use Cases	1
3. Plug-Ins	1
3.1. Identify Plug-Ins	1
3.2. Search Plug-Ins	1
4. Mapserver	1
4.1. Arabic Map Labels	1
5. Documentation	2
5.1. Eclipse	2
5.2. DocBook XSL	2
5.3. Documentation build	2
5.4. CHM Help	2
5.5. PDF Documentation	3
5.6. Arabic Documentation	3
5.6.1. Creating Arabic pdf file	3
5.7. Doxygen	3
6. Subversion	4
6.1. Modules	4
6.2. TortoiseSVN	4
7. Tools	4
7.1. Ant	4
7.2. jEdit	5
7.3. Eclipse	5
7.4. Tidy	5
8. Internationalization	5
9. HTML	5
10. Testing	6
10.1. Manual Testing	6
10.2. Automated Testing	6
11. Procedures	6
11.1. Release Process	6
11.2. Version Number	8
11.3. Adding a Page	8
12. Installer	8
12.1. NSIS	8
13. PostArabic	8
14. Standards	9
14.1. Human Interface	9
14.2. Coding Style	9
14.2.1. Indentation	10
14.2.2. Comments	11
15. Live CD	11
16. Troubleshooting	12
16.1. PostGIS	12
Index	13

List of Examples

- 1. ChangeLog entry 9
- 2. A string variable10
- 3. Echo function10
- 4. Loop indentation10
- 5. If indentation10
- 6. Class indentation11
- 7. Switch indentation11
- 8. Argument separation11

DCMMS Development

1. Introduction

This document describes the *DCMMS* development, including tools and procedures to be applied. It tries to minimize the learning curve for those that want to participate in the *DCMMS* development. Note that this document cannot be comprehensive. Rather than this, it concentrates on the tricky parts and includes references to websites that contain the respective tools and standards.

Additional information is available in *DCMMS API Documentation*, *DCMMS Administration Guide*, *DCMMS User Guide*, various UML diagrams that are part of the *DCMMS* source distribution, on *http://dcmmms.sourceforge.net* and the *dcmmms-devel* mailinglist (<dcmmms-devel@lists.sourceforge.net>).

This document refers to *DCMMS* version 1.1.2+svn .

2. Use Cases

Use cases and workflows for *DCMMS* users.

3. Plug-Ins

Plug-ins allow to extend the functionality of the application in a configurable and transparent way.

3.1. Identify Plug-Ins

Plug-ins for the identify tool of the map page should extend the `IdentifyPlugin` class. See `plugin/imageidentifyplugin.php` for an example.

Identify plug-ins shall read their configuration from `conf/identifyplugin.ini`. Each plug-in should have its individual configuration section.

See the Administration Guide for details on the plug-in configuration.

3.2. Search Plug-Ins

Plug-ins for the find tool at the top of every page should extend `SearchPlugin` class. See `plugin/customersearchplugin.php` for an example.

Search plug-ins shall read their configuration from `conf/searchplugin.ini`. Each plug-in should have its individual configuration section.

See the Administration Guide for details on the plug-in configuration.

4. Mapserver

4.1. Arabic Map Labels

In order to display Arabic map labels, a *PostGIS* database using *UNICODE* should be used. The labels should be *TrueType* labels with a *UNICODE* font that supports Arabic (e.g. Arial).

See the Administration Guide for an example how to load a shape file to such a table.

The Arabic letters have to be shaped, e.g. with the with the *PostArabic* `shape_arabic()` function. See Section 13, "PostArabic" for more details on Arabic shaping.

See also Section 8, “Internationalization”.

5. Documentation

The *DCMMS* documentation is written in *XML DocBook*. DocBook is a very popular format for technical documentation.

5.1. Eclipse



Tip

When editing Docbook files using **Eclipse**, try the *Standalone Structured Source Editor*.



Tip

It is preferable to use Eclipse Version: 3.4.1, this version have the Ant build which helps in building the documentation.

5.2. DocBook XSL

The *DocBook XSL* stylesheets are used to convert the documentation from the *DocBook* format to other formats. *DocBook XSL* is available from <http://www.sagehill.net/docbookxsl/index.html>.

The documentation build file (Section 5.3, “Documentation build”) expects a Docbook XSL installation in `c:\ms4w\lib\docbook-xsl`.

The actual conversion is done by the XSLT processor *saxon*, which is available from <http://saxon.sourceforge.net>. Saxon version 6.5.3 has been used.

The documentation build file (Section 5.3, “Documentation build”) expects `saxon.jar` to be installed in the `c:\ms4w\lib` folder.

5.3. Documentation build

An ANT (Section 7.1, “Ant”) build file (`doc/build.xml`) is available to automate building the documentation in the following formats:

- HTML
- PDF
- CHM

The build file tweaks the generation of the Arabic user guide. You are encouraged to read the build file source.

5.4. CHM Help

It is possible to create HTML Help files (*.chm) from DocBook with the Microsoft HTML Help Workshop.

The HTML help workshop is available under <http://www.microsoft.com/office/ork/xp/appndx/appa06.htm>.

The documentation build file (Section 5.3, “Documentation build”) expects the installation in `c:\Program Files\HTML Help Workshop`.

HTML Help Workshop version 4.74.802 has been used.

5.5. PDF Documentation

PDF Documentation can be generated with *fop*. The Formatting Objects Processor *fop* is available from <http://xmlgraphics.apache.org/fop/>.

The documentation build file (Section 5.3, “Documentation build”) expects the `fop.jar` installation in the `c:\ms4w\lib` folder.

fop version 0.95 has been used.

See also Section 5.2, “DocBook XSL”.

5.6. Arabic Documentation

To build the Arabic documentation, it is currently necessary to shadow the class `com.icl.saxon.charcode.ASCIICharacterSet` with a new version where the function `inCharSet()` always returns `true`.



Tip

`lib\com\icl\ASCIICharacterSet.java` should be compiled like this;

```
java -classpath ..\..\..\..\saxon.jar ASCIICharacterSet.java
```

No PDF is created for the Arabic documentation. Word and PDFCreator can be used to create the PDF from `doc/user/ar/index.html`.

See also Section 5.2, “DocBook XSL”.

5.6.1. Creating Arabic pdf file

Arabic pdf file is not created automatically as the English one because the numbering of sections is viewed from left to right.

So in order to create Arabic pdf file for the user guide the following steps should be done:

- After creating the HTML help files open `Index.html` using Microsoft Word program
- Recreate the table of contents by deleting it and create it again
- Set the header and footer
- Print the Word file to pdf

5.7. Doxygen

Doxygen is used to document the DCMMS API on the PHP level.

Doxygen is available from <http://www.doxygen.org>.

The doxygen configuration file can be found under `doc/development/doxygen/doxygen.cfg`.

In order to generate graphical class representations, you should install Graphviz to `c:\program files\att\graphviz`. Graphviz is available from <http://www.graphviz.org>.

[tp://www.research.att.com/sw/tools/graphviz/](http://www.research.att.com/sw/tools/graphviz/).

Please insert the appropriate comments to document the PHP sources (See Section 14.2, “Coding Style”).

The documentation build file (Section 5.3, “Documentation build”) expects **doxygen.exe** to be installed in `c:\program files\doxygen\bin`.

doxygen version 1.4.6 has been used.

6. Subversion

Subversion is used to manage the *DCMMS* sources.

See Section 11.1, “Release Process” for tagging instructions.

6.1. Modules

The *DCMMS* repository contains the following modules:

- `DCAIS` - an early prototype of the DC Accident Information system.
- `arabic` - the PostgreSQL PostArabic module that allows arabic reshaping in PostgreSQL
- `dcmms` - the DCMMS application itself
- `fuzzystmatch` - the sources of the Fuzzystmatch installer for Cygwin PostgreSQL



Note

As a native PostgreSQL installer for Windows is available, this module is not maintained any more.

- `htdocs` - the DCMMS website
- `postgis` - sources of the Cygwin PostGIS installer



Note

As the native Windows PostgreSQL installer is now including PostGIS, this module is not required any more and thus not maintained. See <http://postgis.refractor.net>

- `sample` - sources of the DCMMS and DCCSS sample data installer.

6.2. TortoiseSVN

See the Sourceforge documentation how to set up SVN access using TortoiseSVN:

http://sourceforge.net/docman/display_doc.php?docid=31165&group_id=1

7. Tools

This section describes a number of tools that will ease the DCMMS development, but are not required for the development.

7.1. Ant

Apache Ant is a build tool that allows to automate many processes.

Ant is available from <http://jakarta.apache.org/ant/>.

Ant is well integrated with the *jEdit* editor - the *jEdit* plugin is called *AntFarm*.

ANT is included in Eclipse.

7.2. jEdit



Tip

Add "-Xmx512m" to your **jEdit** shortcut in order to increase the available heap memory (in this case 512MB).

```
"C:\j2re\bin\javaw.exe" -Xmx512m -jar "C:\jEdit\jedit.jar"
```

Otherwise you'll run out of memory when compiling the documentation.



Tip

The `PHPParserPlugin` is very useful when developing PHP applications.

7.3. Eclipse

Eclipse proved to be a good tool for editing the Arabic User Guide.

Eclipse is freely available from <http://www.eclipse.org>.

The PDT plug-in (<http://www.eclipse.org/pdt/>) is very useful when developing PHP applications.

Eclipse includes ANT.

Eclipse (<http://eclipse.org>) can be used to build the documentation.

See Section 7.3, "Eclipse".

To open the project, select File->Import. Then select General/Existing Projects into Workspace and click Next >. As root folder, select `c:\ms4w\apps\dcmmms`.

To open the Ant view, select Window->Show View->Other.... Then select Ant/Ant.

To build on the documentation, click the Run the selected target button in the Ant view.

7.4. Tidy

The tidy version in the HTML Validator for Firefox is used to validate the generated HTML code.

The HTML Validator for Firefox is available from <http://users.skynet.be/mgueury/mozilla/>.

8. Internationalization

The DCMMS Administration Guide has all the details regarding the translation system using `gettext` and the PL/SQL `dcmmstranslation()` function.

9. HTML

For the onmouseover effects, the following line is important in order to be functional in Internet Explorer:

```
div#links a:hover { border-left: 1px; }
```

10. Testing

10.1. Manual Testing

Manual testing is done using Firefox and Internet Explorer.

The HTML Validator for Firefox is used to check the HTML validity (Section 7.4, “Tidy”).

10.2. Automated Testing

The Canoo Webtest suite is used for testing the application.

Canoo Webtest is freely available from <http://webtest.canoo.com>.

The file containing the tests is `script/test/build.xml`.

To run the tests, follow the steps below:

1. Download `build.zip` for Canoo Webtest version 1.7. Extract the archive to a path without spaces.
2. Add the `bin` folder of the Canoo distribution to the `PATH`.
3. On the command line, change the working directory to `script/test` in the DCMMS sources.
4. Run the following command:

```
runWebtest.bat build.xml
```

The test report will be generated in `script/test/webtest-results/results.html`.



Tip

If you want to generate `WebTest.dtd`, you can use the “`dtd`” target of `script/test/build.xml`. The DTD will allow editors like jEdit to highlight syntax errors in the build file.

11. Procedures

11.1. Release Process

1. Synchronize the configuration files with the default configuration files (`include/dcmconfig.php.default`, `conf/dataobject.ini.default`, `conf/identifyplugin.ini.default`, `conf/sample_*.map`).
2. Update the version number (see Section 11.2, “Version Number”) in `include/header.php`, `include/dcmmspage.php`, `build.xml`, `dcmms.nsi`, `doc/development/doxygen/doxygen.cfg`, `etc/index.html`, `doc/user/*/dcmms_user_guide.xml` and `doc/version.xml`.

3. Update NEWS file and the "What's New?" section in the user guide.
4. Check that the web test is passed.
5. Review Apache's `error.log` after running the web test.
6. Check that the UML diagrams in `doc` are up to date.
7. Update all translations (Arabic, English, French, Italian, Thai). See the Administration Guide for details.
8. Add a new updating section to the administration guide.
9. Check that the documentation build is passed.
10. Create the PDF of the Arabic User Guide from `doc/user/ar/index.html` using Word. The TOC needs to be recreated and header and footer to be added.
11. Build and test the installer. This should be done with the DCMMS CD and a 'virgin' Windows XP installation.
12. Commit everything to the Subversion repository.
13. In the Subversion repository, tag all the files with a tag containing the version information, e.g. *DCMMS_1_0_2*.
14. Delete temporary files.
15. Create a zipfile using `build.xml`. The Installer version number should reflect the version information and an additional packaging number, e.g. `dcmms-1.0.0-1.zip`. The packaging number is increased if packaging problems make it necessary to re-issue a release (e.g. a file is missing in the zipfile). The same version numbering should be applied in the installer.
16. Post the md5sums of the uploaded files in the release notes.
17. Release the files on sourceforge.
18. Download the files and verify md5sums
19. Create a news item on sourceforge.
20. Monitor the discussion forum related to the news item.
21. Update <http://dcmms.sourceforge.net> - this is done automatically with RSS..
22. Update the DCMMS documentation pages on dcmms.sourceforge.net
23. Update the Freshmeat.net record: freshmeat.net
24. Announce the release on `<dcmms-info@lists.sourceforge.net>`.
25. Notify FreeGIS about the new release: <http://www.freegis.org>
26. Announce the new release on <http://www.omsproject.com.jo>
27. Create a CD.
28. Archive CD



Caution

If you distribute *DCMMS* on a CD, make sure that you distribute all the sources. This is required by the DCMMS license (*GPL*).

11.2. Version Number

The DCMMS version number is split in three parts: Major, minor and revision.

DCMMS versions that don't change the data model in the database only increase the revision number (minor feature enhancements, bug fix releases).

For DCMMS releases changing the data model in the database, the minor number is increased.

The major number will be changed in case of large scale technology changes.

11.3. Adding a Page

When adding a page to the application, make sure that the following standards are fulfilled:

1. The page shall be documented in the user guide.
2. A webtest for the new page has to be added.
3. Webtests for the documentation shall be added for all documentation languages.
4. The page shall be fully internationalized

12. Installer

Information on the installer builds of DCMMS and related projects.

12.1. NSIS

The Nullsoft Super Installation System, *NSIS* is used to create the installers used in the DCMMS project.

NSIS is freely available from <http://nsis.sourceforge.net>.

The files `dcmmms.nsi` and `php.nsi` in the `dcmmms` module contain the scripts to create the DCMMS and DCMMS Scripting installers.

The `fuzzystmatch` module contains the `fuzzystmatch.nsi` installer source.

The `sample` module contains the `dcmmms_sample.nsi` installer source.

The `postgis` module contains the `postgis.ini` installer source.

The `arabic` module contains the `postarabic.nsi` installer source.

Follow the instructions in `bin/README.txt` to add the necessary binaries of the GnuWin32 project to the `bin` folder.

To build the DCMMS Scripting installer, several PEAR packages and the PHP/Mapscript extension are required. See `php.nsi` for the details. The PEAR tarballs should be saved in `c:\windows\pear`.

13. PostArabic

The *PostArabic* package provides Arabic shaping functionality for *PostgreSQL*. *PostArabic* is available from <http://dcmmms.sourceforge.net/postarabic.php>.

The package adds the following functions to *PostgreSQL*:

- `text shape_arabic(text)`
- `text unshape_arabic(text)`
- `text str_reverse(text)`
- `bidi_reorder(text)`

See the *arabic* module in the subversion repository and Section 4.1, “Arabic Map Labels” for more information.

14. Standards

Standards-compliance is one of the goals of the *DCMMS* development. This will eventually allow e.g. to support different database backends.

- OpenGIS
- SQL
- UML
- (X)HTML
- CSS
- DocBook
- UNICODE
- Shapefile
- BPMN

14.1. Human Interface

The human interface design tries to follow the *GNOME Human Interface Guidelines* (<http://developer.gnome.org/projects/gup/hig/>).

14.2. Coding Style

Look at the existing codebase and try to code using the same style.

Update the `ChangeLog` for every commit.

Example 1. ChangeLog entry

```
2003-07-29  Steffen Macke <Steffen_Macke@dorsch.com.jo>
           * doc/development/dcmms_development_guide.xml: more
           information on coding style
```

For variable names, use prefixes like `flt`, `str`, `arr` and `date` to indicate the variable type.

Example 2. A string variable

```
$strName = 'test';
```

Variable names should be descriptive, with the exception of simple loop variables like *i,j,k,l,m*

Use a wrap margin of 80 characters. *jEdit* can provide a guiding line for this purpose: Utilities Global Options Editing Wrap Margin

Internationalize all messages visible to the end user. See Section 8, “Internationalization” for the details.

When echoing texts from PHP, omit the brackets:

Example 3. Echo function

```
echo 'test';
```

Single quotes should be used wherever possible for performance reasons.

14.2.1. Indentation**Important**

Use an indent width of 2 characters and a tab width of 8 characters.

Do not use "Soft (emulated with spaces) tabs".

Indent classes, functions, loops and if statements properly. This allows to use *jEdit* folding.

Example 4. Loop indentation

```
foreach($arrNames as $strName) {  
    echo $strName;  
}
```

Example 5. If indentation

```
if($strValue == $strValue2) {  
    echo 'Values are equal.';  
} else {  
    echo 'Values are not equal.';  
}
```

Classes should be properly indented and commented in a way that is compatible with **doxygen**:

Example 6. Class indentation

```
class Test {
    /**
     * A member variable.
     */
    $strTestName;

    /**
     * Test function.
     * @param $strName A name to be passed to the constructor.
     * @return The object.
     */
    function Test($strName) {
        $strTestName = strName;
    }
}
```

Example 7. Switch indentation

```
switch($intValue) {
    case 1:
        $intResult = 2;
        break;
    default:
        $intResult = 1;
        break;
}
```

Separate arguments in function calls by a comma and a space.

Example 8. Argument separation

```
$strMessage = sprintf('This is a %s', 'test');
```

14.2.2. Comments

Use comments to structure the code and make it easier to understand.

Include the copyright header in each source file.

15. Live CD

Information about the KNOPPIX-based DCMMS Live CD.

This is work in progress. If you would like to contribute, please contact the developers.

Configure PHP as a CGI program with the required extensions:

```
./configure --with-gettext --with-gd --with-pgsql
            --enable-force-cgi-redirect --with-zlib
```


`--with-regex=system`

16. Troubleshooting

16.1. PostGIS

Make sure that you use lower case for geometry field names in

- mapserver .map files
- PostGIS function calls like `AddGeometryColumn()`

Index

A

ant, 2
Ant, 4
ANT, 5
Arabic, 1, 8

C

Canoo Webtest, 6
coding style, 9

D

dcmstranslation(), 5
DocBook
 XSL, 2
documentation, 2
 build, 2

E

Eclipse, 2, 5

G

gettext, 5

H

human interface, 9

I

installer, 8
interface
 human, 9
internationalization, 5
introduction, 1

L

Live CD, 11

M

map file, 1
mapserver, 1
module, 4

N

NSIS, 8

P

PostArabic, 8
procedures, 6
project, 5

R

release process, 6

S

sources
 modules, 4
standards, 9
style
 coding, 9
Subversion, 4
 modules, 4
SVN, 4

T

test, 6
 automated, 6
 manual, 6
tools, 4
 TortoiseSVN, 4
TortoiseSVN, 4
translation, 5
troubleshooting, 12

W

webtest, 6