

BMFO Development Guide

Steffen Macke

Abdulqader Jaradat

Baha Faqih

BMFO Development Guide

Steffen Macke
Abdulqader Jaradat
Baha Faqih

Published \$Date: 2006-09-26 21:04:18 +0300 (Tue, 26 Sep 2006) \$\br/>Copyright © 2003, 2004 OMS Project, Northern Governorates Water Authority
Copyright © 2005, 2006 DORSCH Consult

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

General	1
1. Introduction	1
2. Recommended Tools	1
2.1. jEdit	1
2.2. TortoiseSVN	1
2.3. Ant	3
2.4. Tidy	3
Reference	5
1. Documentation	5
1.1. DocBook XSL	5
1.2. Documentation build	5
1.3. CHM Help	5
1.4. PDF Documentation	5
1.5. Arabic Documentation	5
1.6. Doxygen	5
2. Debugging	6
3. Testing	6
3.1. Manual Testing	6
3.2. Automated Testing	6
4. Internationalization	6
4.1. poEdit	7
4.1.1. Settings	7
4.2. Java ResourceBundle	9
5. Database Backends	10
6. Deployment	10
6.1. Installer	10
6.1.1. Client Installer	10
7. Report Creation	11
7.1. iReport	11
7.2. Report Arabization	11
8. Standards	11
8.1. Human Interface	11
8.2. Coding Style	12
8.2.1. Indentation	12
8.2.2. Comments	13
9. Procedures	14
9.1. Release Process	14
9.2. Bugs	14
9.2.1. Bug Tracker	15
10. PEAR	15
Index	16

List of Figures

- 1. TortoiseSVN Context Menu 2
- 2. TortoiseSVN Checkout 3
- 1. poEdit PHP parser settings 8
- 2. Catalog Path 9
- 3. BMFO Deployment 10

List of Examples

- 1. ChangeLog entry 12
- 2. A string variable 12
- 3. Echo function 12
- 4. Loop indentation 12
- 5. If indentation 13
- 6. Class indentation 13
- 7. Switch indentation 13
- 8. Argument separation 13

General

General information for BMFO developers.

1. Introduction

This document describes standards applied in the Bulk Meter Flow and Operations (BMFO) application development.

BMFO is available from <http://bmfo.sourceforge.net>.

Additional information is available in the *BMFO Administration Guide*, the *BMFO User Guide*, the *BMFO API Documentation* and various UML diagrams that are part of the *BMFO* distribution.

The document refers to *BMFO* version 1.2.4 .

2. Recommended Tools

This section describes the software tools that are used in the *BMFO* development.

- jEdit - see Section 2.1
- poEdit - see Section 4.1
- msgmerge - see Section 4.2
- TortoiseSVN - see Section 2.2
- pageant
- WinSCP2
- Visio
- iReport - see Section 7.1

2.1. jEdit

Install the Java Development Kit as described in the *BMFO Administration Guide*.

Install jEdit. jEdit is available from <http://www.jedit.org>

The *AntFarm* plugin was used extensively during the *BMFO* development. It can be obtained through the *jEdit Plugin Manager*



Caution

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

2.2. TortoiseSVN

TortoiseSVN is available from <http://tortoisesvn.tigris.org>.

TortoiseSVN adds a context menu to the Windows Explorer.

Select SVN Checkout from the Explorer context menu (Figure 1).

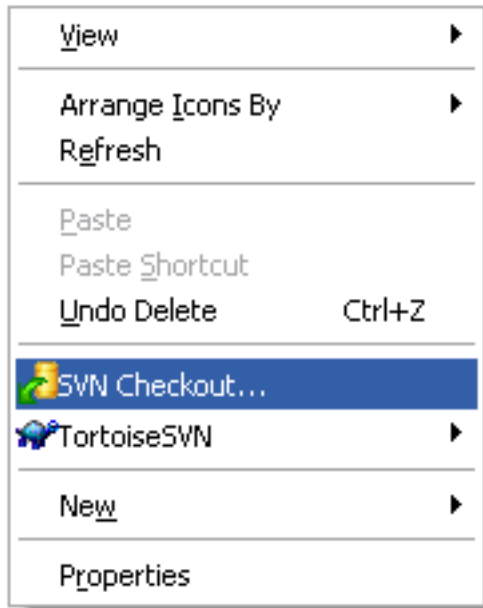


Figure 1. TortoiseSVN Context Menu

As shown in Figure 2, use the URL <https://svn.sourceforge.net/svnroot/bmfo/trunk/bmfo> to check out the BMFO sources.

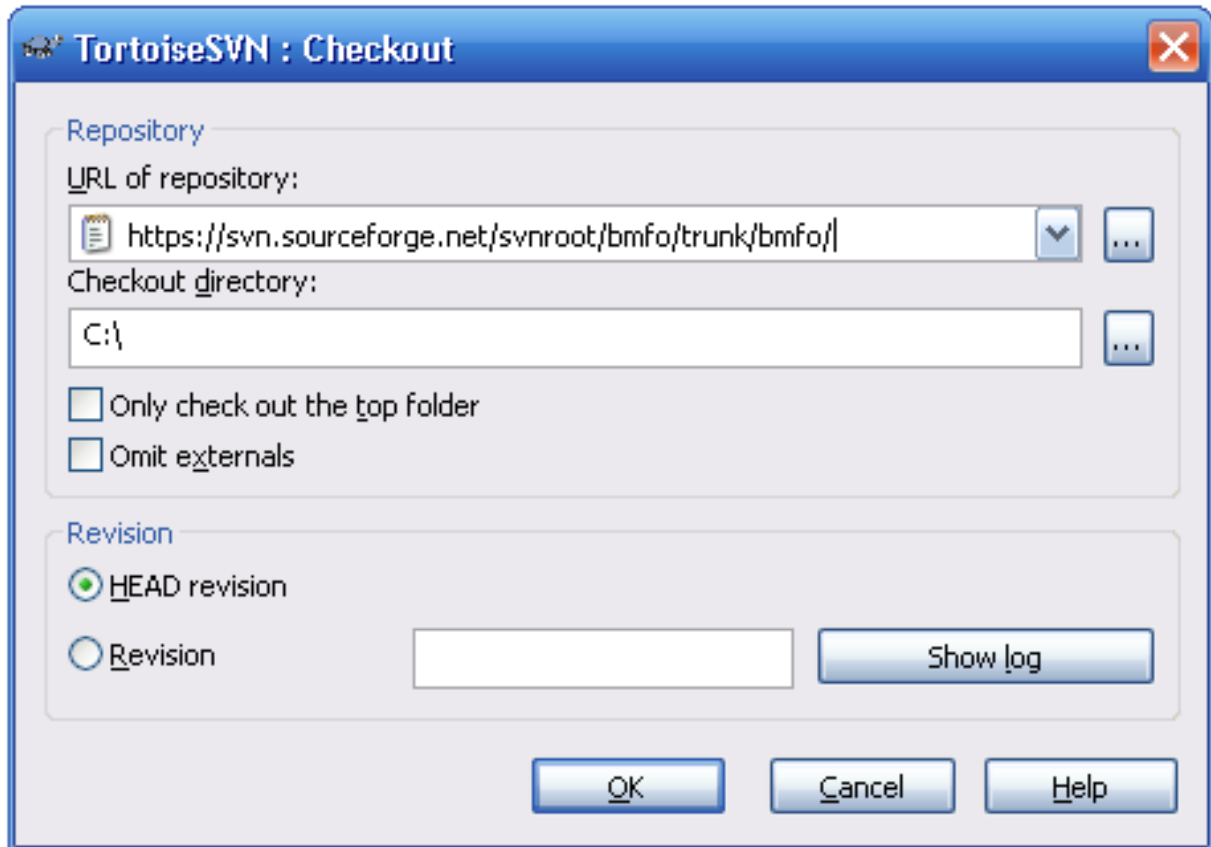


Figure 2. TortoiseSVN Checkout

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

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

2.3. Ant

Apache Ant is a Java-based build tool. It can be used to automate tasks.

1. Download the *Ant* distribution from the internet (<http://ant.apache.org>).
2. Unzip the file to `c:\software`
3. Add `c:\software\jakarta-ant-1.5\bin` to the path.
4. Add a new environment variable, `JAVA_HOME` pointing to your installation of the Java Development Kit, e.g. `c:\program files\java\jdk1.5.0_05`.

The BMFO distribution contains 3 ANT build files:

`build.xml` - used to build the sourcecode distribution
`doc/build.xml` - used to build the documentation from the DocBook sources
`script/test/build.xml` - used to run the Canoo webtest
`servlet/build.xml` - used to build the `bmfo_reports.war` web application archive

2.4. 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/>.

Reference

1. Documentation

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

1.1. 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 actual conversion is done by the XSLT processor *saxon*, which is available from <http://docbook.sourceforge.net>.

1.2. Documentation build

An ant build file (`build.xml`) is available in the `doc` folder to automate building the documentation in the following formats:

- XHTML
- PDF
- CHM

1.3. 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>.

1.4. PDF Documentation

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

1.5. 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`.

The *DocBook XSL* template `language.attribute` in `l10.xsl` should be changed to assign the value `"rtl"` to the attribute `dir` if the language equals `"ar"`.

Arabic PDF: Special processing

1.6. Doxygen

Doxygen is used to document the BMFO 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>.

Please insert the appropriate comments to document the PHP sources.

2. Debugging

Debug output to Apache's `error.log` can be enabled by setting `$GLOBALS['boolDebug']` to true (e.g. in `dbx_connect.php`).

In the source code, debug output should be implemented like in the following example:

```
if ($GLOBALS['boolDebug']) {
    error_log($strQuery);
}
```

3. Testing

3.1. Manual Testing

Manual testing is done using Firefox and Internet Explorer.

The HTML Validator for Firefox is used to check the HTML validity (Section 2.4).

3.2. 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.

4. Internationalization

The *gettext* package is used to internationalize the BMFO application.

Messages that have to be translated but are not included in the PHP sources should be added to `include/i18n.php`. This is the case e.g. for coded values from the database.

4.1. poEdit

To edit the `dcmms.po` files, *poEdit* is recommended. It is freely available from <http://poedit.sourceforge.net>.



Tip

Install poEdit version 1.3.4 or higher.

4.1.1. Settings

In order to extract the translation messages from the PHP sources, the following settings should be applied.

Start **poEdit**.

Select Preferences from the File menu.

Select the Parser tab and click on the New button.

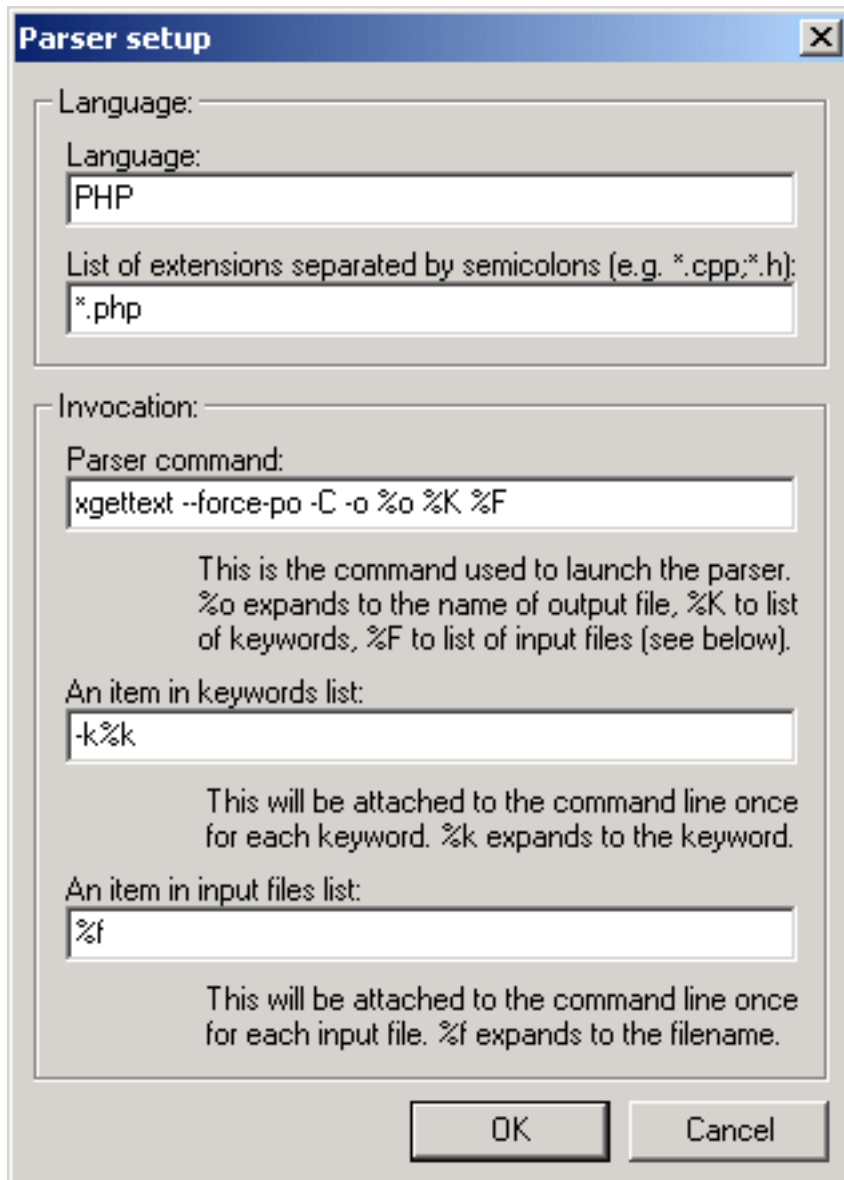


Figure 1. poEdit PHP parser settings

Fill in the settings as shown in Figure 1

Close the dialogs.

Open a dcmsms .po file.

Choose Options... from the Catalog menu.

In the Paths tab, add the path to the *BMFO* source as shown in Figure 2

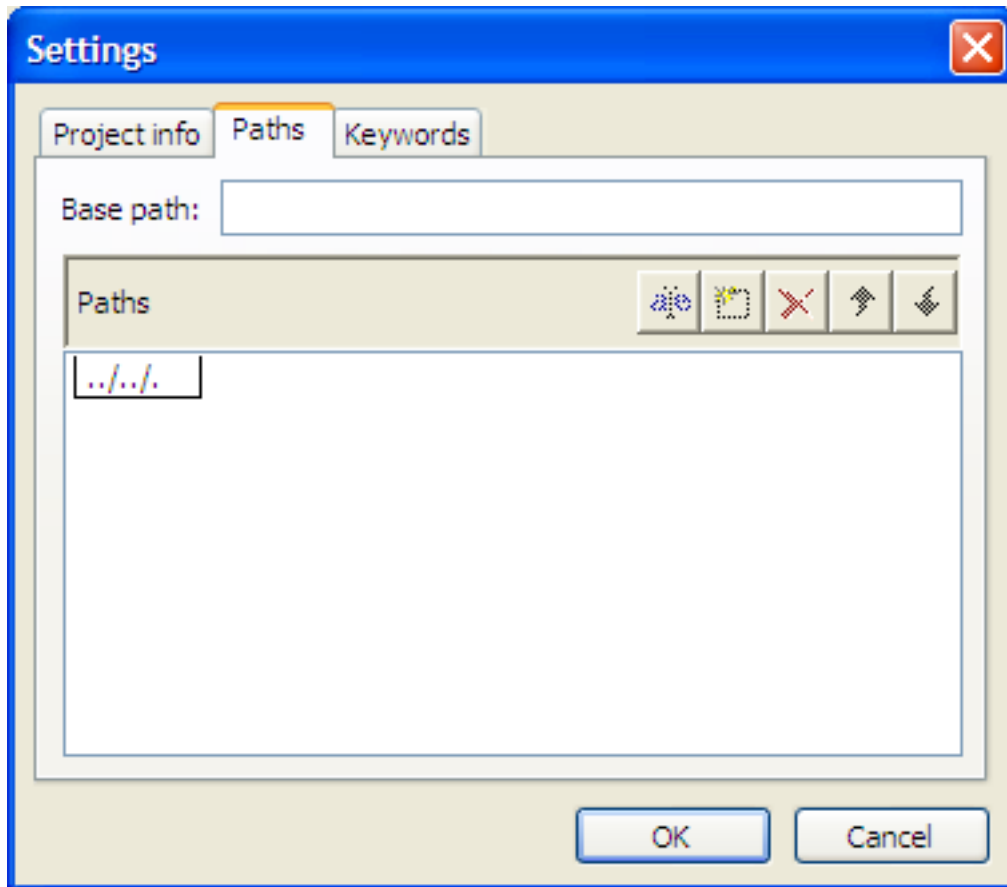


Figure 2. Catalog Path



Caution

Make sure that you enter the actual path to the *BMFO* sources on your system.

Once the settings above have been applied, it is possible to update the catalog by selecting Update from sources from the Catalog menu.

4.2. Java ResourceBundle

The PDF reports are internationalized using Java ResourceBundle. These ResourceBundle can be generated from the *bmfo.po* files using the **msgfmt** command from the *gettext* distribution.

Gettext for Windows is available from <http://gettext.sourceforge.net>.

Create the ResourceBundle for the PDF Reports:

```
msgfmt --java2 -d ..\..\..\servlet\WEB-INF\classes -l ar_AR -r bmfo bmfo.po
```

Note that **javac** must be in your PATH.



Note

If you are using a gettext version < 0.15.0 on Windows you may encounter an error message like the following:

msgfmt: cannot create a temporary directory using template "\msgK5G9pb": Invalid argument

In this case, try to set the environment variable TMPDIR to a temporary directory.

5. Database Backends

BMFO supports different database backends, currently *ORACLE* and *Postgresql*.

All functions that are specific to the different backends have been isolated in `include/dbx_connect.php`. To create `dbx_connect.php`, the appropriate one of the two sample files `dbx_connect_oracle.php` and `dbx_connect_postgresql.php` needs to be copied and adjusted. This is done e.g. by the NSIS installer (Section 6.1).

6. Deployment

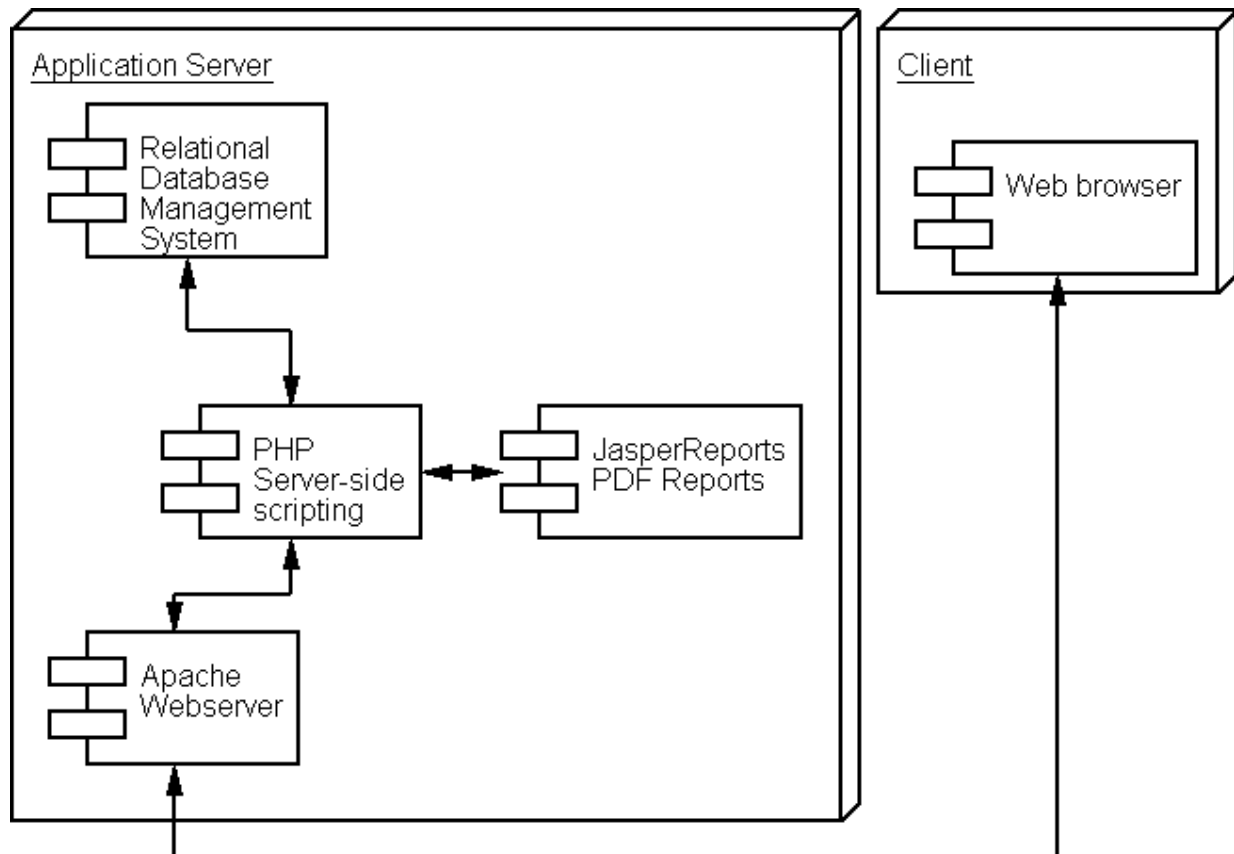


Figure 3. BMFO Deployment

6.1. Installer

To ease the BMFO deployment under Windows, two NSIS installers have been created.

To compile the installers, you need the Nullsoft Super Installation System, which is available from <http://nsis.sourceforge.net>.

6.1.1. Client Installer

The file `bmfo-client.nsi` contains the BMFO Client Installer sources.

Adjust the variable `BMFO_URL` according to the location of your BMFO server:

```
!define BMFO_URL "http://30.35.35.32:81/bmfo/"
```

7. Report Creation

This section describes how to add new PDF reports to *BMFO* using *iReport*.

7.1. iReport

The *iReport* report writer is available from <http://ireport.sourceforge.net>.

7.2. Report Arabization

Use the following font settings for Arabized fields:

- Report font: empty
- Font name: Arial
- PDF Font name: External TTF font...
- True Type font: Arial (arial.ttf)
- PDF Embedded: true
- PDF Encoding: Identity-H (Unicode with horizontal writing)



Caution

Make sure that `arial.ttf` is copied to your *iReport* folder for Arabic support.

8. Standards

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

- SQL
- UML
- XHTML
- CSS
- DocBook
- UNICODE

8.1. Human Interface

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

8.2. 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 4 for the details.

When echoing texts from PHP, use brackets to indicate that `echo()` is a function.

Example 3. Echo function

```
echo("test");
```

Single quotes should be used wherever possible for performance reasons

8.2.1. 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.");
}
```

Example 6. Class indentation

```
class Test {
    // store the name
    $strTestName;

    // constructor
    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");
```

8.2.2. Comments

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

Include the copyright header in each source file.

9. Procedures

Important procedures in the *BMFO* development.

9.1. Release Process

1. Update the version number in `include/header.php`, `doc/version.xml`, `doc/administration/bmfo_administration_guide.xml`, `doc/user/en/bmfo_user_guide.xml`, `build.xml`, `bmfo.nsi`, `bmfo-client.nsi`, `servlet/WEB-INF/web.xml`, `etc/index.html` and `doc/development/doxygen/doxygen.cfg`.
2. Verify that all SQL DDL scripts are running without errors (redirect output to file).
3. Check that the webtest is passed.
4. Review Apache's `error.log` after running the web test.
5. Check that the UML diagrams in `doc/development` are up to date.
6. Update all translations (See Section 4). Don't forget the Java ResourceBundles.
7. Add the necessary information to the Updating section of the administration guide.
8. Check that the documentation build is passed.
9. Update the NEWS file and the "What's New" section in the BMFO user guide.
- 10 In the Subversion repository, copy all files to the appropriate folder under `/tags`. Include the version information, e.g. `/tags/bmfo/BMFO_1_0_8`. This can be done with the TortoiseSVN repository browser.
- 11 Delete temporary files.
- 12 Create a zipfile using `build.xml`. Create the installers.
- 13 Update the documentation on the CD.
- 14 Check the links on the CD.
- 15 Update `etc/index.html`. Create a BMFO CD and test.
- 16 Release the zipfile and installer on sourceforge.
- 17 Upload the updated documentation to Sourceforge.
- 18 Create a news item on sourceforge.
- 19 Update <http://bmfo.sourceforge.net> - this is done automatically with RSS for the main page. Update the docs on the website.
- 20 Update the freshmeat entry (<http://freshmeat.net>).
- 21 Send a release notice to `<bmfo-info@lists.sourceforge.net>`.
- 22 Announce the new release on <http://www.omsproject.com>

9.2. Bugs

For non-trivial bugs, the following approach is recommended:

1. Add a webtest that identifies the bug. Make sure that this test is not passed.
2. Fix the bug
3. Check that all tests are passed.

9.2.1. Bug Tracker

The *Group* attribute of the bug tracker shall be used to determine the version in which the problem will be fixed.



Caution

Don't use the group item to mark the version(s) that are affected by the problem.

10. PEAR

BMFO is currently using the following PEAR classes:

Pager
DataObject
DataObject_FormBuilder

PEAR is the PHP Extensible Application Repository.

It is intended to use more PEAR components in the future.

More information about PEAR can be found under <http://pear.php.net>.

Index

A

Ant, 3
Arabic, 5, 11

B

bmfo.nsi, 10
boolDebug, 6
bug, 14
bug tracker, 15
build.xml, 3

C

Canoo Webtest, 6
ChangeLog, 12
CHM, 5
coding style, 12
configuration
 dbx_connect.php, 10

D

dbx_connect.php, 10
debug, 6
deployment, 10
DocBook, 5
DocBoox
 XSL, 5
documentation
 Arabic, 5
Doxygen, 5

E

error.log, 6

F

files
 bmfo.nsi, 10
 ChangeLog, 12

G

group, 15

H

human interface, 11

I

indentation, 12
installer, 10
Internationalization, 6
 Arabic Reports, 11
 ResourceBundles, 9
introduction, 1
iReport, 11

J

Java
 ResourceBundles, 9
jEdit, 1

N

NSIS, 10

P

PDF, 5
PEAR, 15
poEdit, 7
procedures, 14

R

release process, 14
reports, 11
ResourceBundles, 9

S

standards, 11
style
 coding, 12

T

test, 6
 automated, 6
 manual, 6
tools, 1
 Ant, 3
 Doxygen, 5
 Gettext, 9
 iReport, 11
 jEdit, 1
 NSIS, 10
 poEdit, 7
 TortoiseSVN, 1
TortoiseSVN, 1

W

webtest, 6