



eZ Publish Extension for Oracle(R) database 2.0

eZ Publish Extension Manual

Shyamprakash
information

©1999 – 2010 eZ Systems AS

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 can be downloaded from <http://www.gnu.org/copyleft/fdl.html>.

Corrections and/or suggestions might be sent to info@ez.no.

This PDF file is generated automatically from the online documentation available at <http://doc.ez.no>.

This version was generated on May 20, 2012.

Contents

1	Concepts and basics	5
2	Requirements	6
3	Installation	7
3.1	Automated initialization	13
3.2	Manual initialization	16
3.3	Configuring eZ Publish	19
4	Clustering	25
5	Oracle specific notes	28

Share your information

List of Figures

Shayrre
information

Introduction

The purpose of this manual is to introduce and describe the eZ Publish Extension for Oracle Database (also known as the "ezoracle" extension or "database extension") version 2.0, which makes it possible to use [Oracle](#) as a database server for eZ Publish 4.1. The sections contained within this manual will help you to understand the following issues:

- The purpose of the "ezdb" library
- When the database extension is needed and when it isn't
- How to create a new eZ Publish site running on an Oracle database
- How to migrate an existing eZ Publish site from MySQL to Oracle

Important note

People previously unfamiliar with Oracle should read the "Oracle specific notes (page [28](#))" section first.

Chapter 1

Concepts and basics

The eZ Publish Extension for Oracle Database (also known as the "ezoracle" extension or "database extension") makes it possible to use [Oracle](#) as a database server for eZ Publish.

The communication/interaction between eZ Publish and the database is done via the "ezdb" library, which provides a [database abstraction layer](#). This allows developers to access the database using a generic/unified API that is independent when it comes to different databases and their specific ways of doing things. The "ezdb" library supports MySQL and PostgreSQL databases by default. The database extension contains a database driver implementation (also called the "ezoracle" database handler) that makes it possible for the "ezdb" library to connect to an [Oracle database](#). The extension itself makes use of the [PHP Oracle functions](#). This means that you have to make sure that your PHP installation supports Oracle. Refer to the documentation of PHP for more information.

You need to use the database extension if you wish to do one of the following:

- Create a new eZ Publish site that uses an Oracle database from the beginning.
- Migrate an existing site that uses a MySQL database to Oracle.
- Access an external Oracle database from an eZ Publish site using the ezdb abstraction layer

Note that migration from PostgreSQL is not supported. In addition, once you have migrated from MySQL to Oracle, migrating back to either MySQL or PostgreSQL is not a trivial task, and the database extension does not provide scripts for that.

If your site is running on MySQL/PostgreSQL and you do not plan on migrating to Oracle but only need to execute a few Oracle queries (e.g., you need to fetch some data from the Oracle database for one of your pages), you can choose between the following alternatives:

- Install the database extension and interact with the Oracle database using the wrapper functions provided by the "ezdb" library.
- Do not install the extension and interact with the Oracle database directly using the PHP Oracle functions. Be aware that if the external Oracle database is later replaced by another database solution (MySQL, PostgreSQL, MSSQL, etc.), you will have to rewrite parts of the code that make direct use of the Oracle database functions.

Chapter 2

Requirements

The following list shows the requirements for using the database extension v.2.0.

1. eZ Publish version 4.1 or higher.
2. Oracle Database server software version 9i, 10g or 11g.
3. Oracle Database client software version 9 or higher (Oracle Instant Client is ok). The [SQL*Plus](#) command line tool is recommended.
4. The PHP version that is required for your specific eZ Publish version, with [Oracle support](#). It is strongly recommended to use a recent stable version of the [OCI8](#) extension module for PHP (version 1.2.5 or 1.3.5 at the time of writing). Refer to the [OCI8 changelogs](#) for more information about latest versions of the extension.
5. Proper Oracle environment variables setup (refer to the documentation of the [PHP Oracle functions](#) and the [SQL*Plus tool](#) for more information). Note that eZ Publish 4 requires a UTF-8 database, and thus the [character set component](#) of the NLS_LANG variable can only be set to UTF-8 ("AL32UTF8"). This variable must be set if you use Oracle Database server version prior to 9.2.
6. A user that has the following [system privileges](#) on the Oracle database server:

Privilege	Description
CREATE SESSION	Connect to the database.
CREATE TABLE	Create tables in own schema. Also allows the creation of indexes (including those for integrity constraints) on table in own schema.
CREATE TRIGGER	Create a trigger in own schema.
CREATE SEQUENCE	Create a sequence in own schema.
CREATE PROCEDURE	Create stored procedures, functions, and packages in own schema.

It is recommended that the user also has unlimited quota on the tablespace in which the user schema has been created. This is due to the fact that eZ Publish is a content management system, and therefore it needs to be able to store large amounts of data. The size of the site is usually proportional with the storage needs.

Chapter 3

Installation

This chapter explains how to install version 2.0 of the database extension. The installation typically consists of the following steps:

1. Choose the appropriate installation method
2. Check the requirements
3. Unclusterize images and other files (optional)
4. Make a backup of your current system (optional)
5. Download and unpack the database extension
6. Test connection to an Oracle database server
7. Test Oracle support in PHP
8. Enable the extension
9. Create and initialize an Oracle database for eZ Publish
 - Automated initialization (page [13](#))
 - Manual initialization (page [16](#))
10. Configure eZ Publish
11. Clear the caches
12. Import the unclusterized data to the database (optional)

Note that for the sake of simplicity, this installation manual uses the term "instance name" when referring to [connect identifiers](#). For example, "an instance called X" usually means "an instance that can be connected to using X as a connect identifier", where X is most likely defined in your "tnsnames.ora" file. When using the [easy connect naming method](#), which enables you to connect to an Oracle database without using a "tnsnames.ora" file, a connect identifier could be something like "//dbserver.example.com:1521/ORCL".

1. Choose the appropriate installation method

Before installing the database extension, read the "Concepts and basics (page 5)" section carefully in order to find out about the purpose of the extension and the possible limitations.

There are two possible ways of installing the database extension:

- Migrating an existing eZ Publish site from MySQL to Oracle.
- Creating a new eZ Publish site that uses an Oracle database from the beginning.

The first option assumes that you have eZ Publish already installed and configured. A typical example is when you have an eZ Publish site which is created by the setup wizard using either the "Plain site", "Website Interface" or "eZ Flow" site packages. If the site runs on a MySQL database, you can migrate it to Oracle without breaking the existing functionality.

The other option assumes that an eZ Publish distribution is [downloaded](#) and unpacked to a web-served directory. All further configuration will be done manually. The setup wizard cannot be used in this case. Therefore, if you want to install any of the standard site packages (such as Plain site, eZ Flow, or the Website Interface), you should choose the first option.

If your existing site uses the clustering functionality, you will have to unclusterize the data when migrating from MySQL to Oracle (see step 3 in the instructions below). If you want to continue using clustering after migrating to Oracle, the files should be imported to the new database at a later stage (see step 12). This is because eZ Publish does not use the same database tables for clustering on an Oracle database, and thus the clustering tables cannot be converted automatically when importing the database structure (and data) from MySQL. This will be improved in the future (refer to [this page](#) for more information).

2. Check the requirements

Verify that the requirements (page 6) for the database extension are met.

3. Unclusterize images and other files (optional)

If your existing site stores images, binary files and content-related caches in a MySQL database, you must unclusterize the data before migrating to Oracle. Read the instructions on the "Reverting a cluster setup" documentation page to learn how this can be done.

If your existing site does not use the clustering functionality (or if you want to create a completely new site running on Oracle), skip this step.

4. Make a backup of your current system (optional)

If you are planning to migrate an existing site that uses MySQL to Oracle, you should make a backup of both the eZ Publish directory itself and the MySQL database. Even though the chances for something going wrong are slim, it is still strongly advised to make a backup. Refer to the "Backup" section of the "Upgrading" manual for more information.

The system must be closed to outside access during the backup and installation process.

5. Download and unpack the database extension

The eZ Publish Extension for Oracle Database, which was licensed under the eZ proprietary license before, is now available under the [GPL](#). You can download the latest version [here](#).

Make sure that you copy the downloaded ".zip" package into the "extension" directory of your eZ Publish installation.

The package should be unpacked inside the same directory. When done correctly, an "ezoracle" directory will be created inside the "extension" directory.

On Linux/UNIX, the package can be unpacked with this command:

```
unzip ezoracle-2.0.zip
```

On Windows, you can just unzip the files using the built-in zip features.

At this point, the unpacked files should be available under "extension/ezoracle".

6. Test connection to an Oracle database server

Before doing anything else, make sure you can connect to an Oracle database using an Oracle client. This manual assumes that you are using the [SQL*Plus](#) command-line utility. Assuming that an Oracle instance called "ORCL" is running on your Oracle server and can be accessed using an Oracle account with username "scott" and password "tiger", the shell command for creating a test connection will look like this:

```
sqlplus scott/tiger@ORCL
```

Note that SQL*Plus will retrieve configuration information from your operating system's environment variables (these are listed in the [SQL*Plus documentation](#)). Linux/UNIX users can set their environment manually or by using the "oraenv" utility. On Windows, the Oracle environment variables are stored in the registry.

Example

Linux/UNIX users might not be able to connect using SQL*Plus if the path to the Oracle client libraries directory where the "libsqlplus.so" file is located is not listed in the LD_LIBRARY_PATH environment variable. The following instructions describe how to solve this problem.

1. Check whether the LD_LIBRARY_PATH and ORACLE_HOME environment variables are set:

```
echo $LD_LIBRARY_PATH
echo $ORACLE_HOME
```

2. If the ORACLE_HOME variable is set and points to the Oracle installation directory, set the LD_LIBRARY_PATH variable accordingly:

```
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$ORACLE_HOME/lib
```

Otherwise, locate the "libsqlplus.so" file first and then set the LD_LIBRARY_PATH variable:

```
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:path_to_library
```

Replace "path_to_library" with the full path to the directory where the "libsqlplus.so" file resides (for example "/usr/lib/oracle/10.1.0.4/client/lib").

7. Test Oracle support in PHP

The database extension requires [Oracle support](#) to be enabled in PHP. This means that the [OCI8](#) extension module for PHP must be installed and configured properly. To check if this is taken care of, create a script "test.php" containing the following code:

```
<?php
    phpinfo();
?>
```

and place it in an accessible directory on your web server. Access the script using a browser and look for a section called "oci8" on the browser page.

You can also check this from the command line by running the following command:

```
php -i
```

Note that a restart of the Apache web server might be needed when configuring the OCI8 extension for PHP. In addition, the Oracle environment variables must be set before Apache is restarted, otherwise the [PHP Oracle functions](#) will not work.

8. Enable the extension

To enable the database extension for all of your site accesses, edit the "site.ini.append.php" file located in the "settings/override" directory. Add the following line under the "[Extension-Settings]" section:

```
ActiveExtensions []=ezoracle
```

Note that you'll have to manually create the file and/or the section if they do not exist.

To enable the database extension for only a single site access called "example", edit the "site.ini.append.php" file located in the "settings/siteaccess/example" directory. Add the following line under the "[ExtensionSettings]" section:

```
ActiveAccessExtensions []=ezoracle
```

Note that you'll have to manually create the file and/or the section if they do not exist.

To complete the activation of the extension, the autoload configuration must be regenerated, by running the php script `ezpgenerateautoloads.php`, which can be found in the `bin/php` directory (this is normally done automatically by eZ Publish itself when activating an extension via the administration interface).

9. Create and initialize an Oracle database for eZ Publish

An eZ Publish database must be created on the Oracle server. This means that you will need to initialize the necessary structure and import pre-defined data to the database. You can either do this using the "ora-initialize.sh" script (which attempts to do everything automatically) or manually. Note that the script does not work under Windows. The following text explains both methods.

Automated initialization

Automated initialization is carried out using the "ora-initialize.sh" script. This script is included in the database extension. It is intended to help Linux/UNIX users creating a new eZ Publish site that uses an Oracle database. It is also possible to re-initialize an existing Oracle database that contains eZ Publish related information left after a previous eZ Publish installation. Refer to the "Automated initialization (page 13)" section for more information about this method.

Manual initialization

Manual initialization is a generic method that can be used to either create a new eZ Publish site running on an Oracle database or migrate an existing site from MySQL to Oracle. Both Linux/UNIX and Windows users can perform manual initialization. Refer to the "Manual initialization (page 16)" section for more information about this method.

10. Configure eZ Publish

In order to create a new eZ Publish site that uses an Oracle database, you need to configure file permissions, siteaccesses and languages for your site as described on the "Configuring eZ Publish (page 19)" documentation page.

If you are migrating an existing site from MySQL to Oracle, the only thing you need to do is to change the database settings for your site. To do this, you need to edit the "settings/override/site.ini.append.php" configuration file and make sure that the "[DatabaseSettings]" block contains something that resembles the example below:

```
[DatabaseSettings]
DatabaseImplementation=ezoracle
User=scott
Password=tiger
Database=ORCL
```

The example above assumes that you are using an Oracle database called "ORCL" which can be accessed using "scott" as the username and "tiger" as the password.

11. Clear the caches

It is recommended to clear all eZ Publish caches after installing the database extension. This can be done from within a system shell:

1. Navigate into your eZ Publish directory.
2. Run the script using the following shell command:

```
php bin/php/ezcache.php --clear-all --purge
```

After running the script, make sure that all cache files have been cleared by inspecting the contents of the various cache subdirectories within the "var" directory (typically the "var/cache/" and "var/<name_of_siteaccess>/cache/" directories). If there are any cache files left, remove them manually.

Once the changes outlined above take effect, your eZ Publish installation should be using the Oracle database server. If you log in to the administration interface, access the "Setup" tab and select "System information" from the left menu, you should see "oracle" in the "Database type" section.

12. Import the unclusterized data to the database (optional)

If you didn't unclusterize the files in step 3, skip this step.

If your site was using the clustering functionality on a MySQL database and you want it to continue doing so after migrating to Oracle, you need to set up clustering as described in the "Clustering (page 25)" section. Before doing so, make sure that your site works correctly and you do not experience any problems when running it on an Oracle database.

3.1 Automated initialization

This section describes how you can create a clean eZ Publish database on your Oracle server using the "ora-initialize.sh" script. The script is intended to help those Linux/UNIX users who wish to create a new eZ Publish site running on an Oracle database. (Windows users must do manual initialization (page 16) instead.) It is also possible to re-initialize an Oracle database that contains eZ Publish related information left after previous eZ Publish installations.

The "ora-initialize.sh" script is located in the "extension/ezoracle/scripts" directory and can be executed using the [Bourne Again Shell \(bash\)](#). The following example shows how to run the script.

1. Navigate into the eZ Publish installation directory.
2. Run the script using the following command:

```
./extension/ezoracle/scripts/ora-initialize.sh
```

The script will perform several steps (described below). Note that when the script needs to perform any actions on an Oracle server, it will create a temporary (hidden) file called ".ezoracle_test.php", execute it and then remove it (the temporary file is a script that makes use of the PHP Oracle functions). This is done on every step except the 4-th one. In other words, the script must have write/create privileges in the directory.

Step 1. Checking requirements

The script will check if the following requirements are met:

- PHP CLI is available
- PHP has support for Oracle compiled in
- The eZ Publish installation contains the "ezoracle" extension
- Both the database schema and database data files ("share/db_schema.dba" and "share/db_data.dba") are present in the eZ Publish directory.

The script will also check if the environment variable `ORACLE_HOME`, which points to the Oracle installation directory, is set and the "tnsnames.ora" file is present in the "network/admin" subdirectory. Alternatively, the script will check if the `TNS_ADMIN` variable is set and points to a directory where the "tnsnames.ora" file resides. Depending on whether these variables are set and the "tnsnames.ora" file is present, the script may display a warning to the user before continuing with the remaining tests.

Two other environment variables that will be checked by the script are `LD_LIBRARY_PATH` and `NLS_LANG`. If any of these is not set, a warning will be displayed. If the [character set component](#) of the `NLS_LANG` variable is set to something else than UTF-8 ("AL32UTF8"), the script will display a warning and wait for the user's response, since this configuration is not recommended.

Step 2. Testing connection to an Oracle server (optional)

The script will ask you if this step should be performed or not (skipping is not recommended). If yes, you will have to provide the following information:

Information	Description
Username	The Oracle username that belongs to the Oracle user.
Password	The password that belongs to the Oracle user.
Instance	The name of the Oracle instance. In most cases, it is a simple name specified in your "tnsnames.ora" file.

Step 3. Creating an Oracle user with sufficient privileges (optional)

The script will ask if you wish to create a new Oracle user account. If you already have a user with sufficient privileges, you can skip this step. To create a new user, you will have to provide the following:

- The username and password for the database administrator (or any other Oracle user that has the "CREATE USER" system privilege).
- The username and password for the user that is being created.
- The name of the Oracle instance.
- The name of the default tablespace for the user that is being created (optional).

The script will display the SQL query for creating a new user, ask for confirmation and finally add the user.

Step 4. Connecting to an Oracle server

If you did not create a new user account during step 3, the script will ask you to provide the username/password (not the DB administrator, but the one which the eZ Publish database belongs to) and the name of the Oracle instance. The user must have the required privileges, so that the script can connect to the specified Oracle database and perform the next steps.

Note that this step will be skipped if you created a new user during step 3.

Step 5. Testing user info with Oracle

The script will now attempt to connect to the specified Oracle database using the Oracle account that was specified/created during step 3 or 4.

Step 6. Creating the "md5_digest" function

The database extension requires a custom function called "md5_digest" to be stored in the database. This function returns an MD5 hash (checksum) generated for the provided string input. The script will create this function if the Oracle account that is being used has the "CREATE PROCEDURE" privilege.

Step 7. Creating the "bitor" function

The database extension requires a custom function called "bitor" to be stored in the database. This function returns the result of a bitwise OR operation performed on two numeric arguments. The script will create this function if the Oracle account that is being used has the "CREATE PROCEDURE" privilege.

Step 8. Cleaning up the database

The script will search for existing eZ Publish data in the database and get rid of it (if any).

Step 9. Initializing the database structure

The script will initialize the necessary database structure for eZ Publish (create all the necessary tables, sequences and triggers in the Oracle database) according to the definitions specified in the "share/db_schema.dba" file.

Step 10. Importing pre-defined data to the database

The script will import the pre-defined data from the "share/db_data.dba" file to the database, so that the clean eZ Publish database will be ready to use.

Once the automated initialization has successfully finished, you may continue the installation process by following the remaining steps.

3.2 Manual initialization

This section describes the generic manual initialization method that makes it possible to create a clean eZ Publish database on your Oracle server and/or import data from an existing MySQL database to Oracle. Both Linux/UNIX and Windows users can perform manual initialization.

In order to run eZ Publish on an Oracle database, you must have an Oracle user account with sufficient privileges. If you do not have such an account, proceed with the first step of the instructions below to create a new Oracle user for eZ Publish. When you have such a user, all the remaining steps must be done using this account.

1. Create an Oracle user with sufficient privileges (optional)

In order to create a new Oracle user for eZ Publish, connect to the Oracle instance as the database administrator (or any other Oracle user that has the "CREATE USER" system privilege) and execute the following SQL query:

```
SQL> CREATE USER scott IDENTIFIED BY tiger QUOTA UNLIMITED ON SYSTEM;
GRANT CREATE SESSION TO scott;
GRANT CREATE TABLE TO scott;
GRANT CREATE TRIGGER TO scott;
GRANT CREATE SEQUENCE TO scott;
GRANT CREATE PROCEDURE TO scott;
```

Replace "SYSTEM" with the name of the default tablespace for users if it is configured. (The default tablespace for users can be specified in Oracle Database server software version 10 or higher. Previous versions default to the SYSTEM tablespace if a user is created without the DEFAULT TABLESPACE option.)

This will create an Oracle account with username "scott" and password "tiger". If the user "scott" already exists, you will see the following error message:

```
ORA-01920: user name 'SCOTT' conflicts with another user or role name
```

In this case, you should either choose a different username or make sure that the user with the username "scott" has the necessary privileges.

2. Create the "md5_digest" function

The database extension requires a custom function called "md5_digest" to be stored in the database. This function returns an MD5 hash (checksum) generated for the supplied string input. To add this function, connect to the database using the Oracle account created for eZ Publish and execute the SQL query located in the "extension/ezoracle/sql/md5_digest.sql" file. The following example shows how this can be done assuming that user "scott" can connect to an Oracle instance called "ORCL" and has the "CREATE PROCEDURE" privilege.

1. Navigate into the eZ Publish installation directory.

2. Create the "md5_digest" function using the following shell command:

```
sqlplus scott/tiger@ORCL @extension/ezoracle/sql/md5_digest.sql
```

3. Create the "bitor" function

Another custom function that needs to be added is called "bitor". This function returns the result of a bitwise OR operation performed on two numeric arguments. To add this function, connect to the database using the Oracle account created for eZ Publish and execute the SQL query located in the "extension/ezoracle/sql/bitor.sql" file. The following example shows how this can be done assuming that user "scott" can connect to an Oracle instance called "ORCL" and has the "CREATE PROCEDURE" privilege.

1. Navigate into the eZ Publish installation directory.
2. Create the "bitor" function using the following shell command:

```
sqlplus scott/tiger@ORCL @extension/ezoracle/sql/bitor.sql
```

4. Initialize the database structure and import pre-defined data

It is possible to create a clean eZ Publish database on your Oracle server or import data from an existing MySQL database to Oracle. Follow the instructions given in the corresponding subsection below.

Creating a clean database

In order to create a new eZ Publish site that uses an Oracle database, you will have to initialize the necessary database structure for eZ Publish according to the definitions specified in the "share/db_schema.dba" file and then import pre-defined data from the "share/db_data.dba" file to the database. This can be done by using the "ezsqlinsertschema.php" script located in the "bin/php" directory of your eZ Publish installation. The following example shows how to run this script:

1. Navigate into the eZ Publish installation directory.
2. Initialize the necessary database structure using the following shell command:

```
php bin/php/  
ezsqlinsertschema.php --type=oracle --user=scott --password=tiger share/  
db_schema.dba ORCL
```

The "--clean-existing" option makes it possible to remove eZ Publish data (if it already exists, left-overs from a previous installation):

```
php bin/php/  
ezsqlinsertschema.php --type=oracle --user=scott --password=tiger share/  
db_schema.dba ORCL --clean-existing
```

3. Import the pre-defined data to the database using the following shell command:

```
php bin/php/ezsqlinsertschema.php --type=oracle --user=scott
--password=tiger --schema-file=share/db_schema.dba --insert-types=data share/
db_data.dba ORCL
```

Importing data from MySQL

In order to migrate an existing eZ Publish site from MySQL to Oracle, you will have to import data from an existing MySQL database to Oracle. The following example shows how this can be done assuming that user "root" has password "secret" and can connect to a MySQL database called "mydb" on localhost.

1. Navigate into the "extension/ezoracle/scripts" subdirectory.
2. Run the "mysql2oracle-schema.php" script like this:

```
php mysql2oracle-schema.php mydb:root/secret@localhost > mydump.sql
```

The script will connect to the MySQL database, retrieve the database schema and save it to the "mydump.sql" file in a specific format that is compatible with Oracle.

If the Oracle database that you are going to use already contains some eZ Publish data, the "mysql2oracle-schema.php" script must be run using the "--drop" option, like this:

```
php mysql2oracle-schema.php mydb:root/secret@localhost > mydump.sql --drop
```

3. Import the database schema from "mydump.sql" to the Oracle database:

```
sqlplus scott/tiger@ORCL < mydump.sql
```

If you have used the "--drop" option during the previous step, the "mydump.sql" file will include an appropriate "drop" statement before every "create" instruction, so that the existing elements (if any) will be removed before new ones are created.

4. Run the "mysql2oracle-data.php" script using the following shell command:

```
php mysql2oracle-data.php mydb:root/secret@localhost scott/tiger@ORCL
```

The script will import data from the MySQL database to the Oracle database.

5. Update the sequences in the Oracle database:

```
php ora-update-seqs.php scott/tiger@ORCL
```

5. Continue the installation process

Once the Oracle database is ready, you may continue the installation process by following the remaining steps.

3.3 Configuring eZ Publish

Once you have created a clean eZ Publish database on the Oracle server, you need to configure eZ Publish in order to create a site that runs on the newly created database. Configuration is done manually using the command line interface of the target operating system. The following steps will work on both Linux/UNIX and Windows environments.

File permissions

Windows users can skip this part. If eZ Publish is installed on a Linux/UNIX based system, some of the file permissions need to be changed. There is a shell script that takes care of this. This script must be run, otherwise eZ Publish will not function properly. The script needs to be run from within the eZ Publish directory:

```
$ cd /opt/ezp
$ bin/modfix.sh
```

Replace `"/opt/ezp"` with the full path to the root directory of your eZ Publish installation.

The `modfix` script recursively alters the permission settings of the following directories inside the eZ Publish installation:

- `var/*`
- `settings/*`
- `design/*`
- `autoload/*`

If you know the user and group of the web server, it is recommended to use a different set of permissions:

```
# chown -R user.usergroup var/ settings/ design/ autoload/
# chmod -R 770 var/ settings/ design/ autoload/
```

The `"user.usergroup"` notation must be changed to user and group name that the web server runs as.

Configuring eZ Publish

The `"site.ini.append.php"` configuration file located in the `"settings/override"` directory of your eZ Publish installation must be changed, otherwise eZ Publish will not function properly. This file is the global override for the `site.ini` configuration file. There are a lot of things that need to be configured (database, mail transport system, var directory, etc.). The following text shows a generic example of a configuration that can be used:

```
<?php /* #?ini charset="utf-8"?

[ExtensionSettings]
ActiveExtensions []=ezoracle

[DatabaseSettings]
DatabaseImplementation=ezoracle
User=scott
Password=tiger
Database=ORCL

[FileSettings]
VarDir=var/example

[Session]
SessionNameHandler=custom

[SiteSettings]
DefaultAccess=example
SiteList []
SiteList []=example

[SiteAccessSettings]
CheckValidity=false
AvailableSiteAccessList []
AvailableSiteAccessList []=example
AvailableSiteAccessList []=example_admin
RelatedSiteAccessList []
RelatedSiteAccessList []=example
RelatedSiteAccessList []=example_admin
MatchOrder=host;uri

# Host matching
HostMatchMapItems []=www.example.com;example
HostMatchMapItems []=admin.example.com;example_admin

[InformationCollectionSettings]
EmailReceiver=webmaster@example.com

[MailSettings]
Transport=sendmail
AdminEmail=webmaster@example.com
EmailSender=test@example.com

[RegionalSettings]
Locale=eng-GB
ContentObjectLocale=eng-GB
TextTranslation=disabled
```

```
*/ ?>
```

In the example above, the "AvailableSiteAccessList[]" array located in the "[SiteAccessSettings]" section of this file defines the available siteaccesses called "example" and "example_admin". The "CheckValidity" setting located in the same section should be set to false, so that the setup wizard will not be initiated when trying to access the site. (Note that the setup wizard does not support interaction with an Oracle database server.)

In addition, two siteaccess configurations must be created, a public siteaccess ("example") and an administration siteaccess ("example_admin"). The following subdirectories have to be created in the root of your eZ Publish installation:

- settings/siteaccess/example
- settings/siteaccess/example_admin

Both siteaccesses must have a file called "site.ini.append.php".

The public siteaccess

The following text shows a generic solution for the "example" siteaccess:

```
<?php /* #?ini charset="utf-8"?
```

```
[SiteSettings]
SiteName=Example
SiteURL=www.example.com
LoginPage=embedded
```

```
[SiteAccessSettings]
RequireUserLogin=false
ShowHiddenNodes=false
```

```
[DesignSettings]
SiteDesign=example
```

```
[ContentSettings]
ViewCaching=disabled
```

```
[TemplateSettings]
TemplateCache=disabled
TemplateCompile=disabled
#ShowXHTMLCode=enabled
#Debug=enabled
```

```
[DebugSettings]
DebugOutput=disabled
Debug=inline
#DebugRedirection=enabled
```

```
[RegionalSettings]
SiteLanguageList []
SiteLanguageList []=eng-GB
ShowUntranslatedObjects=disabled

*/ ?>
```

The admin siteaccess

The following text shows a generic solution for the "example_admin" siteaccess:

```
<?php /* #?ini charset="utf-8"?

[SiteSettings]
SiteName=Example
SiteURL=admin.example.com
LoginPage=custom

[SiteAccessSettings]
RequireUserLogin=true
ShowHiddenNodes=true

[DesignSettings]
SiteDesign=admin

[ContentSettings]
CachedViewPreferences [full]=admin_navigation_content=0;
admin_navigation_details=0;admin_navigation_languages=0;
admin_navigation_locations=
0;admin_navigation_relations=0;admin_navigation_roles=0;
admin_navigation_policies=0;admin_navigation_content=0;
admin_navigation_translatio
ns=0;admin_children_viewmode=list;admin_list_limit=1;
admin_edit_show_locations=0;admin_url_list_limit=10;admin_url_view_limit=10;
admin_sec
tion_list_limit=1;admin_orderlist_sortfield=user_name;
admin_orderlist_sortorder=desc;admin_search_stats_limit=1;admin_treemenu=1;
admin_boo
markmenu=1;admin_left_menu_width=13

[DebugSettings]
DebugOutput=enabled
Debug=inline

[RegionalSettings]
SiteLanguageList []
SiteLanguageList []=eng-GB
ShowUntranslatedObjects=enabled
```

```
*/ ?>
```

Note that database settings, mail settings, regional and other settings defined in "settings/override/site.ini.append.php" will be used for each siteaccess regardless of what is specified in the siteaccess settings. In the example above, the "Database=ORCL" is specified under the "[DatabaseSettings]" section of this file so this database will be used for both "example" and "example_admin" siteaccesses. Refer to the "Site management" and "Configuration" documentation pages for more information.

Toolbars in the admin interface

The right section of the administration interface allows you to use the toolbar system. In order to make it work, create the file called "toolbar.ini.append.php" in the "settings/override/example_admin/" directory and make sure that it contains the following lines:

```
<?php /* #?ini charset="utf-8"?

[Toolbar]
AvailableToolBarArray []
AvailableToolBarArray []=admin_right
AvailableToolBarArray []=admin_developer

[Tool]
AvailableToolArray []
AvailableToolArray []=admin_current_user
AvailableToolArray []=admin_bookmarks
AvailableToolArray []=admin_clear_cache
AvailableToolArray []=admin_quick_settings

[Toolbar_admin_right]
Tool []
Tool []=admin_current_user
Tool []=admin_bookmarks

[Toolbar_admin_developer]
Tool []
Tool []=admin_clear_cache
Tool []=admin_quick_settings

*/ ?>
```

Languages

Available languages and their priorities can be controlled per siteaccess using the "SiteLanguageList" configuration setting located under the "[RegionalSettings]" section of the siteaccess "site.ini.append.php" file. If this setting is not specified, the system will use the old "ContentObjectLocale" setting and thus only the default language will be shown. Refer to the "Configuring the site languages" section for more information and examples.

A clean Oracle database created using the "db_data.dba" file contains only one language (British English, eng-GB). All other languages should be added using the "Setup - Languages" part of the administration interface (<http://admin.example.com> in the example above).

Dynamic tree menu

If you have a large site with many nodes, it is strongly recommended to enable the "Dynamic" switch for your administration site access. This will make the left tree menu in the administration interface work much faster and decrease the usage of network bandwidth.

Administrator's login and password

The following username and password are specified in the "db_data.dba" file and can be used for logging in to the administration interface.

- Username: admin
- Password: publish

It is strongly recommended to change this password as soon as possible. Note that if you need another username for site administrator, you can create a new administrator user, log in as this user and remove the old one.

Once the configuration is done, you can finish the installation process by following the last steps.

Chapter 4

Clustering

You can configure eZ Publish to store images, binary files and content-related caches in an Oracle database. This is typically useful if you want to run a site on a cluster of servers (refer to the "Clustering" section in the "Features" chapter of the eZ Publish Technical Manual for more information).

The following instructions explain how this can be done, assuming that you already have an eZ Publish site running on an Oracle database. The procedure consists of the following steps:

1. Clearing the caches (optional)
2. Modifying the "file.ini" settings
3. Creating a new script for serving images
4. Creating new database structures
5. Importing files to the database
6. Compiling the templates (optional)
7. Updating the Apache configuration
8. Restarting Apache and testing the site
9. Removing the imported files from the filesystem

These steps are very similar to the ones that are used when configuring eZ Publish to use clustering on a MySQL database, with a few differences in the first four steps, as described below.

1. Clearing the caches (optional)

It is recommended (but not required) to clear all eZ Publish caches before enabling the clustering functionality. This can be done by running the "bin/php/ezcache.php" script as shown below:

```
php bin/php/ezcache.php --clear-all --purge
```

After running the script, make sure that all cache files have been cleared by inspecting the contents of the various cache subdirectories within the "var" directory (typically the "var/cache/" and "var/<name_of_siteaccess>/cache/" directories). If there are any cache files left, remove them manually.

2. Modifying the "file.ini" settings

Add the following lines to an override for the "file.ini" configuration file ("settings/override/file.ini.append.php" or "settings/siteaccess/ezwebin_site/file.ini.append.php" where "ezwebin_site" is the name of your siteaccess):

```
[ClusteringSettings]
FileHandler=ezdb
DBBackend=oracle
DBHost=
DBPort=
DBSocket=
DBName=ORCL
DBUser=scott
DBPassword=tiger
DBChunkSize=65535
ExtensionDirectories []=ezoracle
```

Replace "ORCL", "scott" and "tiger" with the actual data:

- Name of the Oracle instance (usually a simple name specified in your "tnsnames.ora" file)
- User name
- Password

In most cases these values will be the same as the "Database", "User" and "Password" settings specified in the [DatabaseSettings] block of your "site.ini.append.php" configuration file.

The "DBHost", "DBPort" and "DBSocket" settings can be left empty, since they are not used when running eZ Publish on an Oracle database.

Setting the "FileHandler" configuration directive to "ezdb" will instruct eZ Publish to use the specified database for storing images, binary files and content-related caches. The "DBBackend" setting specifies the back-end to be used by the "ezdb" file handler. The "ExtensionDirectories" configuration array specifies the extension directories where eZ Publish should search for additional cluster file handlers and/or back-ends. The settings above will instruct eZ Publish to use the back-end located in "extension/ezoracle/clusterfilehandlers/dbbackends/oracle.php".

The "DBChunkSize" setting determines the size of the blocks (in bytes) into which files are split when they are fetched from the database.

3. Creating a new script for serving images

All images (except design images) will be served by PHP. Apache will be instructed to use a specific PHP script called "index_cluster.php" for handling images. The script must include "index_image.php" along with a collection of configuration settings. This trick makes the serving of images faster because the system does not have to read the configuration from the database. Create the "index_cluster.php" inside the eZ Publish root directory and make sure that it contains the following lines:

```
<?php
define( 'STORAGE_BACKEND',      'oracle'          );
define( 'STORAGE_USER',        'scott'            );
define( 'STORAGE_PASS',        'tiger'           );
define( 'STORAGE_DB',          'ORCL'            );
define( 'STORAGE_CHUNK_SIZE',  65535             );

include_once( 'index_image.php' );
?>
```

Make sure you specify the same database settings as indicated in the "[ClusteringSettings]" block in your "file.ini.append.php" configuration file.

You must also copy the "index_image_oracle.php" file from the "extension/ezoracle" directory to the root directory of your eZ Publish installation.

4. Creating new database structures

You have to create some structures in the database manually. You'll find the needed definitions in the "oracle.php" file located in the "extension/ezoracle/clusterfilehandlers/dbbackends" subdirectory. The definitions are placed inside comment blocks in the beginning of the file.

When using the SQL*Plus command line tool, be sure not to miss the ending slash ("/") in the last line of the PL/SQL code copied from the "oracle.php" file.

5. Importing files to the database

Refer to the "Clustering - Setting it up" documentation page for details on steps 5-9.

Chapter 5

Oracle specific notes

This section describes some Oracle specific terms and concepts mentioned in the documentation of the eZ Publish Extension for Oracle Database. Refer to the [official documentation of Oracle](#) for more information.

An Oracle database is a collection of specific files (control files, redo log files, data files, undo files and temp files). Every running Oracle database is associated with an Oracle instance. When a database is started on a database server, Oracle allocates a memory area (SGA) and starts one or more background processes (SMON, PMON and others). This combination of processes and shared memory is called an *Oracle instance*. An instance only exists when it is running. A database is normally managed by one and only one instance, unless the [Oracle Real Application Clusters \(RAC\)](#) technology is used.

You can also look at an Oracle database as a *service* that stores and retrieves data for clients. In a single instance environment (non-RAC), the words "service" and "instance" are often used as synonyms; the target instance to connect to is identified by a database [service name](#).

In order to connect to an Oracle instance, a client needs a connect descriptor, which is a specially formatted description of the destination for a network connection. In most cases, such a descriptor specifies the protocol, host, port and service name that should be used. It is possible to define simple names that map to connect descriptors. For example, these can be defined in a "tnsnames.ora" file stored on a client. Users initiate a connect request by passing a username and password along with a simple name using the following format:

```
username/password@simple_name
```

In the latest versions of Oracle, the [easy connect naming method](#) can be used. It allows a client to connect to a database server with a connect string for a simple TCP/IP address, consisting of a host name and optional port and service name:

```
username/password@[//]host[:port] [/service_name]
```

A *schema* is a named collection of objects (tables, indexes, procedures, etc.) associated with a user account. The word "schema" can also be used to refer to an Oracle account. Oracle database installations usually come with a default schema that can be accessed using "scott" as the username and "tiger" as the password.

An Oracle database is divided into one or more logical storage units called *tablespaces*. Tables and indexes are created within a particular tablespace. The primary tablespace in any

database is the SYSTEM tablespace, which Oracle creates automatically when the database is created. This tablespace is always online when the database is open. You can assign each user a tablespace *quota* in order to limit the amount of space allocated for storage of the user's objects within the specified tablespace. By default, a user has no quota on any tablespace in the database. If the user has the privilege to create a schema object, you must assign a quota to allow the user to create objects.

*SQL*Plus* is an Oracle command line tool for running SQL and PL/SQL commands. This utility is often used by database administrators and developers to interact with the Oracle database. You can interact with SQL*Plus directly through the command line or through scripts. Refer to the [documentation of SQL*Plus](#) for more information.

Share
information