Database Extension 1.4

eZ Publish Extension Manual

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 February 18, 2009.

Contents

List of Figures

Introduction

The purpose of this manual is to introduce and describe the eZ Publish Extension for Oracle Database (also known as ”ezoracle” extension or ”database extension”), which makes it possible to use Oracle as a database server for eZ Publish. 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 30)” section first.

Chapter 1

Concepts and basics

The eZ Publish Extension for Oracle Database (also known as ”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. Please 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.

Please note that migration from PostgreSQL is not supported. In addition, once you have migrated from MySQL to Oracle, it is not possible to migrate back to either MySQL or PostgreSQL.

If your site is running on MySQL/PostgreSQL and you do not plan 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.

1. eZ Publish version 3.7 or 3.8.
2. Oracle 8 or 9 (client and server software, SQL*Plus command line tool). Note that Oracle 10 is not recommended for this version of the database extension.
3. PHP 4.4 (preferably the latest version of the 4.4 branch) with Oracle support.
4. Proper Oracle environment variables setup (please refer to the PHP Oracle functions documentation for more information).
5. 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.
ALTER ANY TABLE	Alter any table in any schema and compile any view in any schema.
ALTER ANY TRIGGER	Enable, disable, or compile any trigger in any schema.
ALTER ANY SEQUENCE	Alter any sequence in any schema.
ALTER ANY PROCEDURE	Compile any stored procedure, function, or

	package in any schema.
DROP ANY TABLE	Drop or truncate any table in any schema.
DROP ANY TRIGGER	Drop any trigger in any schema.
DROP ANY SEQUENCE	Drop any sequence in any schema.
DROP ANY PROCEDURE	Drop any stored procedure, function, or package in any 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, 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 1.4 of the database extension. The installation typically consists of the following steps:

1. Make a backup of your current system (optional)
2. Check the requirements
3. Download the database extension
4. Copy the downloaded package
5. Unpack the downloaded package
6. Enable the extension
7. Test connection to an Oracle database server
8. Create and initialize an Oracle database for eZ Publish
    - Automated initialization (page 12)
    - Manual initialization (page 15)
9. Configure eZ Publish
10. Restart Apache

1. 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. Please refer to the ”Backup” section of the ”Upgrading” manual for more information.

2. Check the requirements

Please verify that the requirements (page 6) for being able to use the database extension are met.

3. Download the database extension

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

4. Copy the downloaded package

Copy the downloaded ”tar.gz” package into the ”extension” directory of your eZ Publish installation.

5. Unpack the downloaded package

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

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

```
$ tar -zxvf ezoracle-extension-1.4.tar.gz
```

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. Enable the extension

To enable the database extension for all of your siteaccesses, edit the ”site.ini.append.php” file located in the ”settings/override” directory. Add the following line under the ”[ExtensionSettings]” section:

```
ActiveExtensions[]=ezoracle
```

Please 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 siteaccess 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
```

Please note that you’ll have to manually create the file and/or the section if they do not exist.

7. 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
```

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

Example

UNIX/Linux 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 $LD_LIBRARY_PATH. The following shell commands can be used to display and set this environment variable:

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

8. 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. Please note that the script does not work under Windows. Also, you might have problems running the script on UNIX/Linux if you’re using the Oracle Instant Client. The following text reveals 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 UNIX/Linux 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. Please refer to the ”Automated initialization (page 12)” 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 UNIX/Linux and Windows users can perform manual initialization. Please refer to the ”Manual initialization (page 15)” section for more information about this method.

9. Configure eZ Publish

In order to create a new eZ Publish site that uses an Oracle database, you need to configure file permissions, siteaccesses, Unicode support and languages for your site as described on the ”Configuring eZ Publish (page 18)” documentation page.

If you are migrating an existing site from MySQL to Oracle, the only thing you need to do is changing the database settings for your site. Simply open the ”settings/override/site.ini.append.php” configuration file and change the ”[DatabaseSettings]” block to 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.

10. Restart Apache

A restart of the Apache web server is required in order for the changes outlined in the previous steps to take effect. Once it has been restarted, your eZ Publish installation should be using the Oracle database server (if you log in to the admin interface, access the ”Setup” tab and select ”System information” from the left menu, you should see ”oracle” in the ”Database type” section). Please note that the Oracle environment variables must be set before Apache is restarted, otherwise the PHP Oracle functions will not work.

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 UNIX/Linux users who wish to create a new eZ Publish site running on an Oracle database (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).

Please note that if you are using the Oracle Instant Client, the ”ora-initialize.sh” script might not work correctly because the ”ORACLE_HOME” environment variable is probably not in use and the script will not find the ”$ORACLE_HOME/network/admin/tnsnames.ora” file. In this case, you can manually set this environment variable and copy the ”tnsnames.ora” file to the ”$ORACLE_HOME/network/admin/” directory before running the script. The following example demonstrates how this can be done.

1. Make sure that the ”ORACLE_HOME” environment variable is not set:

```
echo $ORACLE_HOME
```

2. Locate the ”tnsnames.ora” file (in most cases it can be found in ”/etc/tnsnames.ora” or ”~/.tnsnames.ora”).
3. Create the ”~/network/admin/” directory and copy the ”tnsnames.ora” file there. (If there is a dot in the beginning of a filename, remove it.)
4. Set the ”ORACLE_HOME” environment variable:

```
export ORACLE_HOME=path_to_your_home_directory
```

The following example shows how to run the ”ora-initialize.sh” 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). Please 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
- The environment variable $ORACLE_HOME, which points to the Oracle installation directory, is set and the ”tnsnames.ora” file is present it the ”network/admin” subdirectory
- PHP has support for Oracle compiled in
- The eZ Publish installation contains the ”ezoracle” extension
- Both database schema and database data files (”share/db_schema.dba” and ”share/db_data.dba”) are present in the eZ Publish directory.

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, these names (connect strings) are specified in the ”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 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.

Please 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 ”CREATE PROCEDURE” privilege.

Step 7. Cleaning up the database

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

Step 8. 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 instructions specified in the ”share/db_schema.dba” file.

Step 9. 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 UNIX/Linux and Windows users can perform manual initialization. The following instructions explain how it can be done.

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

You will not be able to run eZ Publish on an Oracle database unless you have an Oracle user with sufficient privileges. If you do not have such a user, connect to the Oracle database 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;
 GRANT ALTER ANY TABLE     TO scott;
 GRANT ALTER ANY TRIGGER   TO scott;
 GRANT ALTER ANY SEQUENCE  TO scott;
 GRANT ALTER ANY PROCEDURE TO scott;
 GRANT DROP  ANY TABLE     TO scott;
 GRANT DROP  ANY TRIGGER   TO scott;
 GRANT DROP  ANY SEQUENCE  TO scott;
 GRANT DROP  ANY PROCEDURE TO scott;
```

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 chose 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 an Oracle account that has the ”CREATE PROCEDURE” privilege 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. 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. Please 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 instructions 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:

```
./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):

```
./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:

```
./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:

```
./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:

```
./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:

```
./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:

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

4. 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 UNIX/Linux and Windows environments.

File permissions

Windows users can skip this part. If eZ Publish is installed on a UNIX/Linux based system, some of the file permissions need to be changed. There exists 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/ezpublish/3.x
$ bin/modfix.sh
```

Replace the ”/opt/ezpublish/3.x” by 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/*

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/
# chmod -R 770 var/
```

Change the ”user.usergroup” to the user and group used by the web server (Apache).

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="iso-8859-1"?

[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
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. (Please 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="iso-8859-1"?

[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

# the following settings can only be used in eZ Publish 3.8 and later versions:
[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="iso-8859-1"?

[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
kmarkmenu=1;admin_left_menu_width=13
```

```
[DebugSettings]
DebugOutput=enabled
Debug=inline

# the following settings can only be used in eZ Publish 3.8 and later versions:
[RegionalSettings]
SiteLanguageList[]
SiteLanguageList[]=eng-GB
ShowUntranslatedObjects=enabled

*/ ?>
```

Please 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. Please refer to the ”Site management” and ”Configuration” sections of the ”Concepts and basics” chapter for more information.

Unicode support

If your Oracle database uses Unicode character set and PHP is compiled with multibyte string support, create the file called ”i18n.ini.append.php” in the ”settings/override/” directory of your eZ Publish installation and make sure that it contains the following lines:

```
<?php /* #?ini charset="iso-8859-1"?

[CharacterSettings]
Charset=utf-8
MBStringExtension=enabled

*/ ?>
```

Issue the following SQL query to determine the character set encoding used by your database:

```
SELECT value FROM nls_database_parameters WHERE parameter = ’NLS_CHARACTERSET’
```

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="iso-8859-1"?

[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

A clean Oracle database created using the ”db_data.dba” file contains only one language (British English, eng-GB). Please note that if you are using eZ Publish 3.7 (or earlier versions), the default primary language will be set to eng-GB as well and objects will have to be created in the primary language before they could be translated to additional languages.

All other languages should be added using ”Setup - Languages” (or ”Setup - Translations” in 3.7 and earlier versions) in the administration interface (http://admin.example.com in the example above).

Default username and password

The following username/password combination is specified in the ”db_data.dba” file. Use it to log into the administration interface.

- Username: admin
- Password: publish

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

Chapter 4

Upgrading

Every new version of the eZ Publish Extension for Oracle Database comes with database upgrade scripts of the following two types:

- Scripts that make it possible for your current eZ Publish installation to use the latest version of the extension. These are located in the ”update/database/ezoracle/” directory of the database extension.
- Scripts that make it possible to upgrade eZ Publish to the latest stable version. These are located in the ”update/database/ezpublish/” directory of the extension.

Please note that as long as your site is already running on an Oracle database, the database extension makes it possible to upgrade the entire eZ Publish installation and not just the extension itself .

Consistency checks

Make sure that your eZ Publish site is in a consistent state. The administration interface allows you to quickly check whether the current installation is in a consistent state or not. This can be done using the ”Upgrade check” section of the ”Setup” part. Please refer to the eZ Publish Upgrading Manual for more information about consistency checks.

Backup

Make sure that you have a working backup of the site before you do the actual upgrade. You should create a copy of the entire eZ Publish directory and the database. Please refer to the Oracle backup and recovery documentation page and/or Oracle Backup and recovery FAQ for information about backing up the database.

Upgrading the database extension

Use this method if you only want to upgrade the extension itself but not eZ Publish. In this case, the upgrade procedure consists of the following steps:

1. Upgrading the extension files
2. Upgrading the database
3. Clearing the INI cache
4. Restarting Apache (optional)

Step 1. Upgrading the extension files

Download the distribution package from http://ez.no/download/add_ons/database_extension and unpack it inside the ”extension” directory. (Please refer to the ”Installation (page 8)” section for more information about how to download and unpack the eZ Publish Extension for Oracle Database.)

Step 2. Upgrading the database

When a new version of the database extension is released, the database structure is usually changed/updated. In order for your eZ Publish installation to use the latest version of the database extension, you need to run the corresponding database upgrade script(s). These are located in the ”extension/ezoracle/update/database/ezoracle/” directory. The following example demonstrates how this can be done when upgrading the eZ Publish Extension for Oracle Database from version 1.3 to 1.4.

Example

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”, you need to do the following:

1. Navigate into the ”extension/ezoracle/” directory.
2. Run the script using the following shell command:

```
$ sqlplus scott/tiger@ORCL @update/database/ezoracle/1.4/dbupdate-1.3-1.4.sql
```

Step 3. Clearing the INI cache

Upgrading the database extension requires the INI file caches to be cleared. This should be done from within a system shell:

1. Navigate into the eZ Publish directory.
2. Run the clear cache script:

```
$ bin/shell/clearcache.sh --clear-ini
```

Step 4. Restarting Apache (optional)

You may need to restart the Apache web server after upgrading, especially if your site is using a PHP accelerator. Restarting Apache is usually a good idea when it comes to getting rid of strange results and doubts. Please note that the Oracle environment variables must be set before Apache is restarted, otherwise the PHP Oracle functions will not work (please refer to the PHP Oracle functions documentation for more information).

Upgrading eZ Publish

Use this method if you are already running the latest version of the database extension and wish to upgrade your eZ Publish installation to a newer version. In this case, the upgrade procedure consists of the following steps:

1. Upgrade the eZ Publish distribution files
2. Upgrade the database
3. Run the system upgrade scripts
4. Update the system configuration
5. Clear the caches

Note that the upgrade procedure is basically the same as described in the eZ Publish Upgrading Manual. (Please refer to the ”Upgrading from 3.a.b to 3.x.y” section or read the upgrade instructions for your particular version of eZ Publish.) The only difference is related to the database upgrade under step 2. The database upgrade scripts are located inside the ”ezoracle/update/database/ezpublish” directory (starting from 3.5, because eZ Publish versions prior to 3.5 can not be run on an Oracle database server).

The following example demonstrates how to upgrade the database 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”.

Example

Let’s say that you are using version 1.4 of the database extension and that you wish to upgrade eZ Publish from version 3.7.0 to 3.8.1. This means that you should follow the upgrade procedure described above. If you remembered to copy the ”extension/ezoracle” directory while upgrading the eZ Publish distribution files during step 1, the ”ezoracle/update/database/ezpublish/3.8” directory will contain the following database upgrade scripts:

- dbupdate-3.6.0-to-3.8.0.sql
- dbupdate-3.8.0-to-3.8.1.sql

During step 2, you need to upgrade the 3.7.0 database to 3.8.1 by running these two scripts (in the specified order) following the official upgrade path. In other words, you need to do the following:

1. Navigate into the ”extension/ezoracle/” directory.
2. Run the following shell commands in the same order as specified:

```
$ sqlplus scott/tiger@ORCL @update/database/ezpublish/3.8/
dbupdate-3.6.0-to-3.8.0.sql
$ sqlplus scott/tiger@ORCL @update/database/ezpublish/3.8/
dbupdate-3.8.0-to-3.8.1.sql
```

The remaining three steps are described in the ”Upgrading to 3.8” section of the eZ Publish Upgrading Manual.

Upgrading eZ Publish and the database extension

This method is a combination of the two upgrade procedures described above. Use it if you wish to upgrade both eZ Publish and the database extension at the same time. The upgrade procedure consists of the following steps:

1. Upgrade the eZ Publish distribution files and the database extension files
2. Upgrade the database
3. Run the system upgrade scripts
4. Update the system configuration
5. Clear the caches
6. Restart Apache (optional)

Example

Let’s say that you are using eZ Publish 3.7.0 together with version 1.3 of the database extension and wish to upgrade to versions 3.8.1 and 1.4 accordingly. In this case, there is no need to copy the ”extension/ezoracle/” directory from the existing installation to the eZ Publish 3.8.1 directory when upgrading the distribution files during step 1. Instead, you need to download version 1.4 of the database extension from http://ez.no/download/add_ons/database_extension and unpack it inside the ”extension” directory.

During step 2, you need to run the database upgrade scripts from both ”extension/ezoracle/update/database/ezoracle/1.4/” and ”extension/ezoracle/update/database/ezpublish/3.8/” subdirectories.

Steps 3-5 are described in the ”Upgrading to 3.8” section of the eZ Publish Upgrading Manual. Step 6 is already described in the ”Upgrading the database extension” subsection.

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. Please refer to the official documentation of Oracle for more information.

An Oracle database is a collection of specific files residing on disk (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 while it is up and running. A database is normally managed by one and only one instance.

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. The ”tnsnames.ora” file can be used to define net service names (often referred to as ”connect strings”) as aliases for connect descriptors. Users initiate a connect request by passing a username and password along with a connect string using the following format: ”username/password@connect_string”.

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. SQL*Plus can be used interactively or through scripts.

Oracle stores a single hash for each username/password combination rather than just the password itself. The passwords are not encrypted but hashed (digested) using the realm information as the salt. This results in a fixed length string. Oracle provides a set of MD5 generation functions included in the DBMS_OBFUSCATION_TOOLKIT PL/SQL package.

share your information