



Share your information

Database Extension 1.5

eZ Publish Extension Manual

©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	Oracle specific notes	8

List of Figures

Share
information

Introduction

The eZ Publish Extension for Oracle Database makes it possible to use [Oracle](#) as a database server for eZ Publish. Version 1.5 of the extension is a bug fix release, with the same requirements, and installation and upgrade procedures as version 1.4.

Refer to the documentation of the eZ Publish Extension for Oracle Database version 1.4 for the installation manual and upgrade instructions.

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

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.

Source
Your
information

Shyare

information