TECHNICAL TRACK
JPA driver – extended database support for OpenCms

Contents

1. What is JPA - Java Persistence API
2. Apache OpenJPA project
3. JPA driver integration
4. Benefits of JPA driver
5. Configuration parameters
6. Installing OpenCms with JPA driver
7. Setting JPA driver on top of existing OpenCms installation
8. Performance tests with MySql and PostgreSQL
9. OpenCms modules and JPA integration
10. Using OpenCms to cut costs in times of recession

1. What is JPA

- the API, defined in the `javax.persistence` package
- the Java Persistence Query Language
- object/relational metadata

Existing JPA implementations

- Hibernate (LGPL)
- TopLink Essentials (CDDL)
- EclipseLink (Eclipse Public License)
- OpenJPA (Apache License)
- JPOX (Apache License)

2. Apache OpenJPA project

JPA1 - certified
JPA2 - on going work for implementation of new features

Project's features :

- Good community support
- Good documentation
- Good performance
- Wide range of supported databases
- OpenJPA is highly configurable
- Simple jar files dependencies
- Distributed persistence (Slice plugin)
- Flexible open-source license

3.JPA driver

The driver consists of:

- org.opencms.db.jpa.CmsXxxDriver
- persistence classes (org.opencms.db.jpa.persistence.*)
- query.property file
- code for EntityManager pool configuration
- Extended CmsDbContext implementation
- Patch for the installer

3.JPA driver comparison

Differences between JPA and SQL driver

Feature	SQL	JPA
Transactions scope	Within methods of the driver	Within methods of CmsSecurityManager
Query language	SQL	JPQL
Number of driver implementation	Equal to number of supported databases	Single implementation for all databases
Pool for EntityManager instances	NA	implemented
Data cache	NA	Strong cache system
Configuration file	opencms.properties	persistence.xml

4. Benefits of JPA driver

- Simplifying of database layer
- Decreased time for testing of db layer
- Increased number of supported databases
- Use of data caching at application side

Status of JPA driver for OpenCms 7

Test	Number of tested databases
Supported databases by OpenJPA	18
Installation	9
JUnit tests	4 (Oracle, MySql, PostgreSQL, Firebird)

5.Configuration parameters

Here is following configuration parameter groups:

- OpenCms
- Connection management
- OpenJPA cache
- Data dictionary

OpenCms parameters

Parameter	Default value
opencms.jpa.EntityManagerPoolSize	300

Connection management

Parameter	Default value
DriverClassName	NA
Url	NA
Username	NA
Password	NA
MaxActive	20
maxIdle	20
initialSize	20
poolPreparedStatements	true
openjpa.ConnectionRetainMode	on-demand

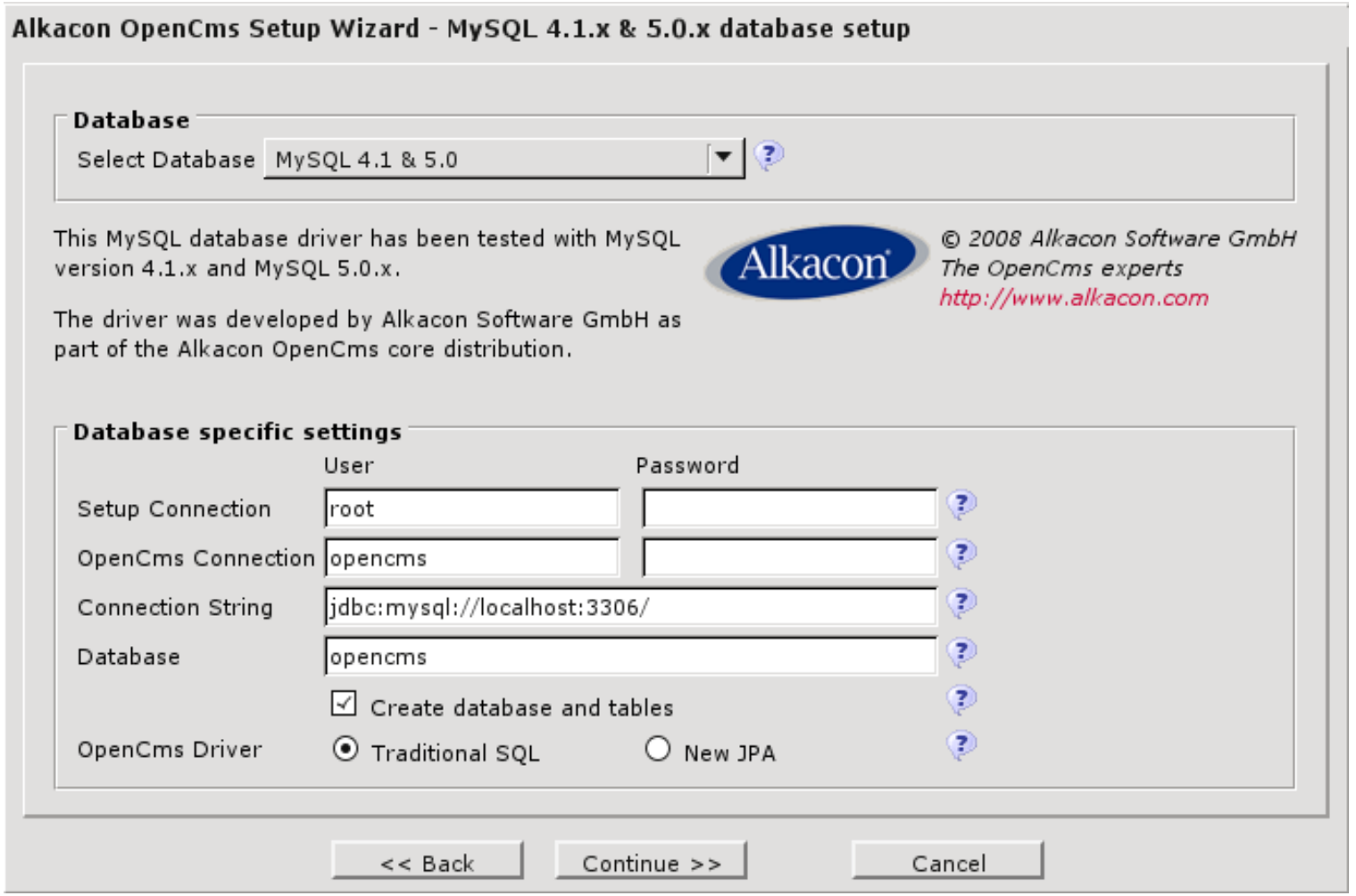
Cache parameters

Parameter	Default value
openjpa.RemoteCommitProvider	sjvm
openjpa.DataCache	true(CacheSize=5000, SoftReferenceSize=0)
openjpa.QueryCache	CacheSize=500, SoftReferenceSize=0
openjpa.QueryCompilationCache	true
openjpa.jdbc.QuerySQLCache	true

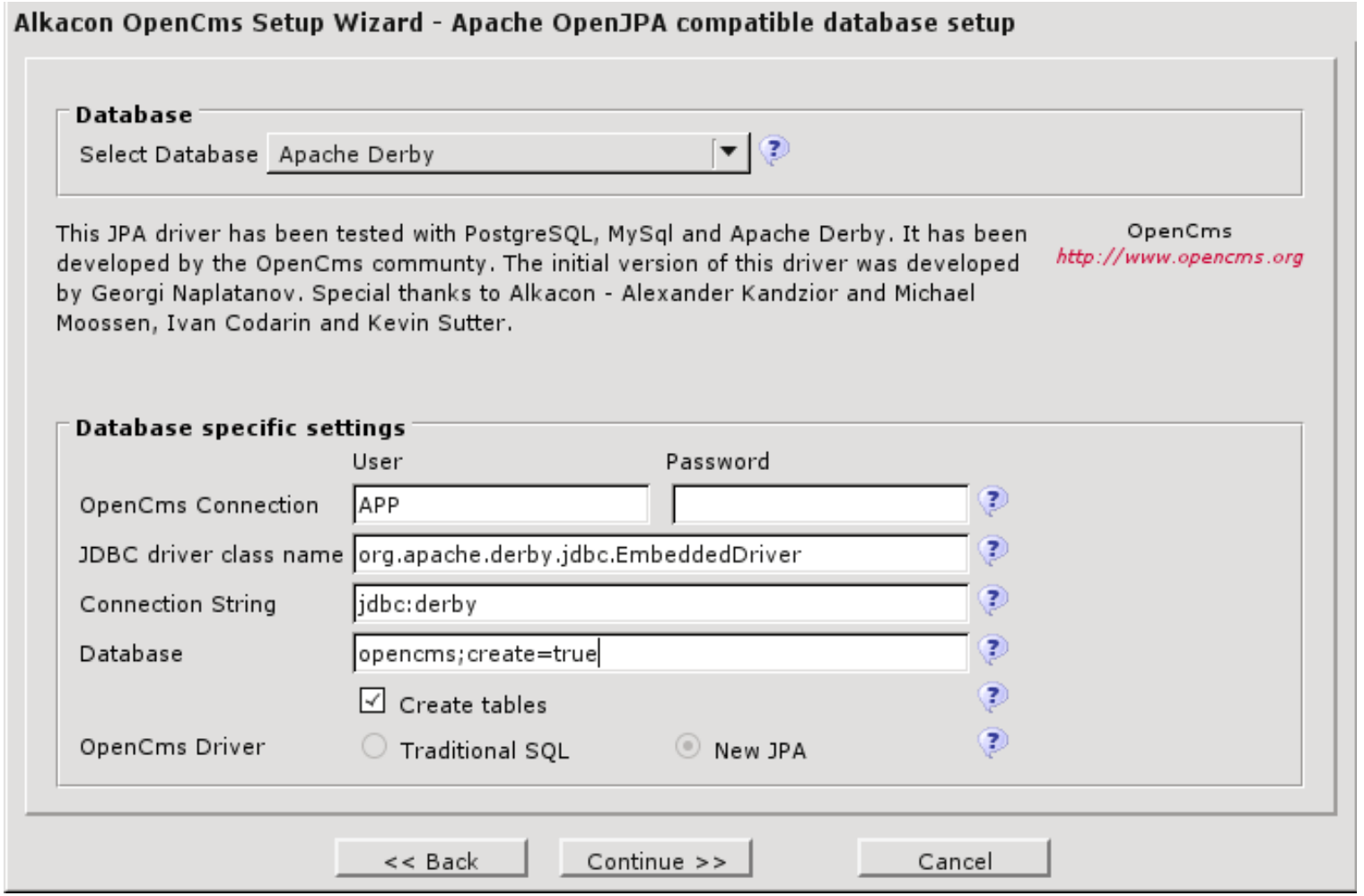
Data dictionary

Parameter	Value
openjpa.jdbc.DBDictionary	mysql(SupportsSubselect=true)
openjpa.jdbc.DBDictionary	oracle(maxEmbeddedBlobSize=-1, maxEmbeddedClobSize=-1)
openjpa.jdbc.DBDictionary	JoinSyntax=traditional, SupportsUniqueConstraints=false

Installation - MySql



Installation – Apache Derby



7. Setting JPA driver on existing inst.

- copy persistence.xml file and set up connection pool
- copy opencms.jar
- copy OpenJPA's jar files
- set up JPA driver in the opencms.properties file – all CmsXxxDriver classes should be from the org.opencms.db.jpa package
- remove “not null” constraint from cms_contents.file_content and cms_offline_contents.file_content for Oracle and DB2 databases

Performance test - configuration

Component	Description
CPU	AMD Athlon(tm) 64 X2 Dual Core Processor 4200+
RAM	4 GB DDR-2 800
OS	Debian GNU/Linux, Lenny AMD64
JDK	Sun 1.6.0_14
JSP/Servlet container	Apache Tomcat 6.20
Databases	PostgreSQL 7.3.7, MySQL 5.0.51a
Pool size	1000 database connections for SQL driver and 1000 EntityManager instances with 100 db connections for JPA driver

Test

#ab -c 200 -n 4000 http://testserver:8080/opencms/opencms/demo_en/intro.html

8. Performance tests result

RDBMS	Driver	Pool size	Time (s)	Performance
PostgreSQL	SQL	1000	31.93	
PostgreSQL	JPA	1000/100	33.13	+3.75 %
MySQL	SQL	1000	33.70	
MySQL	JPA	1000/100	25.97	+29.75%

9.OpenCms modules and JPA integration

- Use OpenJPA's reverse mapping tool
- Choose which persistence unit to use

Benefits of using OpenCms's persistence unit

- Use of connection pools for EntityManager and database connections

Benefits of using separate persistence unit

- different JPA implementation can be used
- different database can be used

10. Using OpenCms to cuts costs of recession

- there is no need to re-qualification of your database administrators, OpenJPA supports 18 different databases
- OpenJPA supports most used open-source databases
- JPA driver use smaller number of database connections than SQL drivers