



## **RFP 77 - Relational Database Service**

Open Distribution, Draft

7 Pages

### **Abstract**

This document addresses the need to specify an common access method to databases, sharing of connections and related concerns.

Copyright © Jayway 2007.

All company, brand and product names contained within this document may be trademarks that are the sole property of the respective owners.

The above notice must be included on all copies of this document that are made.

---

# 0 Document Information

---

---

## Table of Contents

<b>0 Document Information.....</b>	<b>2</b>
Table of Contents.....	2
Terminology and Document Conventions.....	2
Revision History.....	2
<b>1 Introduction.....</b>	<b>3</b>
<b>2 Application Domain.....</b>	<b>3</b>
Terminology + Abbreviations.....	3
<b>3 Problem Description.....</b>	<b>4</b>
<b>4 Use Cases.....</b>	<b>4</b>
<b>5 Requirements.....</b>	<b>5</b>
<b>6 Document Support.....</b>	<b>6</b>
References.....	6
Author's Address.....	6
End of Document.....	6

---

## 0.1 Terminology and Document Conventions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY" and "OPTIONAL" in this document are to be interpreted as described in 6.1.

Source code is shown in this typeface.

## 0.2 Revision History

The last named individual in this history is currently responsible for this document.

Revision	Date	Comments
Initial	Feb 22 2007	Initial Draft Niclas Hedhman, Jayway, <a href="mailto:niclas.hedhman@jayway.net">niclas.hedhman@jayway.net</a>
Changed	July 21 2007	Incorporated use cases and requirements from RFP 66 – DataSource Service, which was covering mobile concerns Stuart McCulloch, Jayway, <a href="mailto:stuart.mcculloch@jayway.net">stuart.mcculloch@jayway.net</a>

---

# 1 Introduction

---

This document will only cover connections between OSGi applications and Relational Database Management Systems (RDBMS), and not others such as hierarchical, network-oriented or object-oriented databases. It covers both enterprise and mobile concerns relating to connectivity.

In fact, it solely deals with the re-use of previous work around the JDBC specification and to make JDBC fit well into the OSGi environment.

---

# 2 Application Domain

---

Both mobile and enterprise applications have similar requirements with respect to locating database resources.

Open Distribution, Draft

June 22, 2007

Enterprise applications are typically database intensive, and this RFP considers the circumstances around database connectivity for OSGi applications in the enterprise sector.

Enterprise application bundles are often independent of each other, and many applications are residing on many OSGi frameworks, potentially 100s or even 1000s in some form of cluster.

It also deals with mobile devices – as their data consumption grows, there is a need for new storage mechanisms. JSR 169 (JDBC optional package for CDC/Foundation Profile) allows Java applications to access relational databases using SQL. JSR 169 doesn't include a driver manager for managing different database drivers from different vendors, but this should not mean the mobile solution should differ from the enterprise solution, as the core API is still JDBC.

This document is mainly centered around the JDBC Specifications [3], which are implemented by endless number of vendors and we must ensure compatibility with that legacy.

---

## 2.1 Terminology + Abbreviations

RDBMS – Relation Database Management System

JDBC – Java Database Connectivity

---

# 3 Problem Description

---

Many applications are database centric, especially in the enterprise sector. OSGi doesn't provide any unified way to get hold of connections to databases, and each bundle or application is left to define its own way of doing this.

Many existing technologies leverages relational database systems to make them more object oriented friendly. Hibernate, JDO and JPA are examples of this. However, if we create a JDO implementation bundle, we also need to find a way to provide the JDBC driver to that bundle from another bundle, as it is not reasonable to assume that the JDO implementation is delivered with all possible JDBC drivers.

RDBMS' are also limited in the number of concurrent connections that they can handle. And since establishing the connection is time consuming, applications must keep the connections open and re-use them to the highest degree possible. In OSGi, this poses the problem that bundles don't know of each others existence, and if we

Open Distribution, Draft

June 22, 2007

deploy a large number of bundles that each uses its own connection, the maximum connection number will be reached fairly quickly.

If we look at OSGi deployments in clustered environments, the problem becomes more amplified, and connections becomes a very limited resources.

OSGi is a promising technology in both Enterprise and Mobile sectors, and one of the cornerstones are database connectivity. There is no specifications in OSGi yet,

---

## 4 Use Cases

---

Mohammad Isac Niclas bin Abdullah is writing a JDO implementation, which uses JDBC drivers to establish a connection to a persistent relational store. Mohammad does not want to ship any JDBC drivers inside the JDO implementation bundle and needs to use drivers provided by the user of his bundle. Mohammad also needs that the user of his bundle can change the database connection at any time, without stopping the JDO bundle. Preferably it should be possible for the user to migrate the database from one vendor to another, change the JDBC driver type without the stopping and re-resolving of the JDO bundle. The settings for the JDBC driver, normally provided either in the connection string, in system properties or in a separate properties file, should additionally be possible to configure from the Configuration Admin service of OSGi.

Edward Yakop is writing portlets to be sold to customers. Each portlet is independent of every other portlet, and the work is based on legacy code where the portlets do their own JDBC connections to the database. The system therefore has a connection manager bundle which hands out pooled connections on demand to the portlets. Each customer has their own database vendor and Edward wants to be able to offer that the customer can use any database that has a standard OSGi compliant JDBC driver available.

Stuart McCulloch is writing a mobile application that accesses a remote database using the JSR 169 optional JDBC package – he wants to provide his application as a series of bundles running on a standard mobile OSGi platform. The phone vendor wants to support JSR 169 while maintaining isolation and avoiding any impact on core phone services.

## 5 Requirements

---

The solution MAY leverage the Service registry in OSGi.

The solution MUST allow multiple connections to many databases, incl usage of JDBC drivers of different versions, within and amongst the bundles.

The solution SHOULD provide a way where the JDBC driver is installed as a bundle, and the connections are instantiated declaratively in runtime.

The solution MUST provide a mechanism to associate a connection description in the client with a actual connection of the provider.

The solution MUST support JDBC 2.0 and later.

The solution SHOULD support all types of JDBC drivers, Type 1, 2, 3 & 4.

The solution MUST include a mechanism for sharing of database connections among bundles.

The solution MAY require co-operative sharing of connections, i.e. acquire/release cycles.

The solutions MUST allow the number of connections between the OSGi framework and the database to be limited to a relatively small number, and be manageable in runtime.

The solution MUST allow for monitoring of the activity on the database connections.

The solution MUST ensure compatibility with existing JDBC drivers.

The solution SHOULD allow for inter-framework connection pools.

The solution SHOULD allow implementations to prevent the direct use of the JDBC DriverManager.

The solution MUST not rely on the existence of the JDBC DriverManager class.

The solution SHOULD provide full XA support for **any** resource.

The solution MUST support XA in JDBC implementations that supports XA.

The solution MUST support both mobile and standard OSGi frameworks.

---

# 6 Document Support

---

---

## 6.1 References

- [1]. Bradner, S., Key words for use in RFCs to Indicate Requirement Levels, RFC2119, March 1997.
- [2]. Software Requirements & Specifications. Michael Jackson. ISBN 0-201-87712-0
- [3]. JSR-54 "JDBC 3.0 Specification", JSR-221 "JDBC 4.0 Specification", JSR-991 "JDBC 2.1 Errata Sheet", JSR-169 "JDBC Optional Package for CDC/Foundation Profile"

---

## 6.2 Author's Address

Name	Niclas Hedhman
Company	Jayway
Address	B-7-6, Megan Avenue 1, 50450 Kuala Lumpur, Malaysia
Voice	pstn:+46 40 621 8888, pstn:+1 360 226-8029, mobile: +60 12 2277 953, sip:2000@sip.codedragons.com
e-mail	niclas.hedhman@jayway.net

---

## 6.3 End of Document