Open Source Requirements Management Tool

Installation Manual

Version 1.8

June 2019

Revision History

Date	Version	Description	Author
06-20-2019	1.8	Updated Version 1.8	Suchita Arunan, Choo
05-05-2019	1.7	Updated Version 1.7	Suchita Arunan, Choo
02-08-2019	1.6	1.6 Rebuild	Anton Tim, Suchita Arunan
03-26-2007	1.5	1.5 Update	Aron Smith
12-20-2006	1.4	1.4 Update	Aron Smith
10-10-2006	1.3	PostgreSQL/ MySQL updates	Thomas, Andre Felipe Machado
10-10-2006	1.3	1.3 Update	Aron Smith
8-27-2006	1.2	1.2 Update	Aron Smith
6-21-2006	1.1	1.1 Update	Aron Smith
05-11-2006	1.0	Initial Draft	Aron Smith

Table of Contents

1. Introduction

Open Source Requirements Management Tool (OSRMT) is a highly configurable, free open source solution for defining and managing Requirements for software development. This is an easy to install and easy to use solution with capabilities to document all aspects of software development life cycle (SDLC).

This solution was initially built in 2006. However no further development happened for many years. In 2019, the redevelopment was initiated by Alan Clifford and a new version (Version 1.6) was released in Feb 2019 at GitHub. The latest Version is 1.8 that was released in June 2019. This can be accessed through **https://github.com/osrmt/osrmt**

1.1 **Purpose of this Document:**

The purpose of this document is to provide technical and operational assistance for the installation of OSRMT application.

It also provides guide lines on installing and using various databases with OSRMT.

1.2 Who can use OSRMT?

a) Single user using their work station
b) A group of users independently working, but willing to share a common database
c) A corporation or organization with multiple users including multi location users.

1.3 Installation Options

OSRMT Application can be installed in any of the below scenarios

2 Tier Architecture where the Application connects directly to a Database and has the following installation options

Single user installation where the Database and Application resides in the same computer

Multi user installation where the Database is centralised and Application resides in computer of every user.

3 Tier Architecture where the Application connects to the Database through the J2EE server and has the following installation options

Multi user installation with multiple servers and a central database.

Multi user installation **with** Central database and web browser clients

2. Installation

2.1 Install Java Standard Edition, version 8 or higher

Download and Install Java runtime environment (JRE).

2.2 Setting up 2 Tier Application (Desktop Application)

Install Client software – WINDOWS

Download desktop application

Once the download is completed, extract the archive ***OSRMT.desktop_v1.8.zip***

From here you can use desktop application as is, proceed to step **5**. If you want to set up specific database for it, follow step **4**.

By default app uses MS Access DB which doesn't need any additional configurations, just proceed to step ***d.a*****,** but if you want to use one of the following DBMS: MySQL, Oracle, Postgres - go inside extracted folder and follow **Install Database** steps before first start.

After database is set up (if it was needed) and configured:

for Windows users: execute run.bat

for Linux and MacOS users:

execute command "chmod +x run.sh"

execute ./run.sh

TIPS

For the 2 Tier single user installation

OSRMT server is not required
Separate Java installation may not be required

2.3 Setting up 3 Tier Application (Windows) (Server Application with Web GUI)

2.3.1 Install Software

Download the client software from OSRMT Web application

Once the download is completed, extract the file ***OSRMT.web_v1.8.zip***

From here you can use web application as is, proceed to step **5**. If you want to set up specific database for it, follow step **4**.

By default, OSRMT application uses MS Access DB which doesn't need any additional configurations. You can directly proceed with the installation.

However, if you want to use a different database, you can use ***MySQL, MS SQL, Oracle or Postgres.***

To use any of the above data bases, please go inside extracted folder and follow **Install Database** steps before starting the installation.

After database is set up and configured please proceed as follows:

enter jboss-4.0.3/bin

for Windows users: execute run.bat

for Linux and MacOS users:

execute command "chmod +x run.sh"

execute ./run.sh

Once the application is deployed in JBoss server on 8080 port, it will be accessible locally using the url: http://localhost:8080/osrmt

TIP

In case you want to use another port

Go to jboss-4.0.3/server/default/deploy/jbossweb-tomcat55.sar and update port in file server.xml.

Server restart is needed after this action.

2.4 Complicated installation option - Desktop Application with centralized server.

If you are interested in Desktop Application with centralized server installation follow manual steps in **5. Install the j2ee osrmt server for a 3 tier configuration**

3. Create and Install new database

This section is optional if you do want to use some specific database from the following list: MySQL, MS SQL, Oracle, PostgreSQL

Follow the vendor specific instructions to create a database and a login.

The login consists of a username and password which allows access to the database.

Click Here to Install MySQL

Click here to Install Oracle

Click here to Install PostgreSQL

Click here to Install MS SQL

3.1 Setup database environment

Follow the vendor specific instructions to create a database and a login.

The login consists of a username and password which allows access to the database.

Attention!	Desktop application Database scripts location **<path_to_extracted_app>/schema** and for Web application it is **<path_to_extracted_app>/dbscripts/schema**. Further will be called <**schema_path**>

3.2 Mysql

3.2.1 Download and install MySQL 5.0 or higher

Download and install MySQL 5.0 or higher from the following Link

https://dev.mysql.com/downloads/

Install MySQL (make sure to remember root user and password to be able to connect to DB after installation finishes)

3.2.1 Create OSRMT Database

Start →Program files → MySQL → MySQL Command line client

create database osrmt;

show databases;

3.2.2 Create Schema

From the command line execute the create schema and create view script source <path>

connect osrmt;

connect osrmt;

*>source <**schema_path**>\mysql_create_schema.sql*

*>source <**schema_path**>\mysql_create_view.sql*

ignore warning message when dropping view ERROR 1051 (42S02): Unknown table 'osrmt.artifactdetail'

3.2.3 Create User

create user osrmt identified by 'osrmt';

grant all on osrmt. to osrmt;*

3.3 Oracle

3.3.1 Download and install Oracle

Install Oracle Express and create a database instance. For more details refer to Database Express Edition Getting Started Guide

3.3.2 Create User/Schema

sqlplus system/manager

create user osrmt identified by osrmt

default tablespace <some tablespace your choice> temporary tablespace <some tablespace your choice>;

grant dba to osrmt;

3.3.3 Create Schema Tables and Views

From the command line execute the create schema and create view scripts

sqlplus osrmt/osrmt

*>@<**schema_path**>\oracle_create_schema.sql*

*>@<**schema_path**>\dbscripts\schema\oracle_create_view.sql*

TIPS

See the appendix for sample vendor instructions on databases against which OSRMT has been validated.

3.3.4 Create database tables and indexes

Execute the shipped <db>_create_schema.sql script which can be found in the schema directory e.g.

<schema_path>\oracle_create_schema.sql

<schema_path>/mysql_create_schema.sql etc.

3.3.5 Create database views

Execute the shipped <db>_create_view.sql script which can be found in the client schema directory e.g.

<schema_path>\sqlserver_create_view.sql etc.

TIPS

Ignore any warnings which states that " ***the view did not previously exist"***

3.4 SQL Server Express Database Setup Notes

3.4.1 Download and install SQL Express

Install Microsoft SQL Server 2005 Express Edition

Install SQL Server Management Studio Express.

Open Microsoft SQL Server Management Studio Express and connect to your SQL Server instance.

3.4.2 Create OSRMT Database

Right click on the Database folder and select New Database…

Enter OSRMT as the database name and press OK

Set ansi padding = false

Set ansi nulls default = true

3.4.3 Create Login

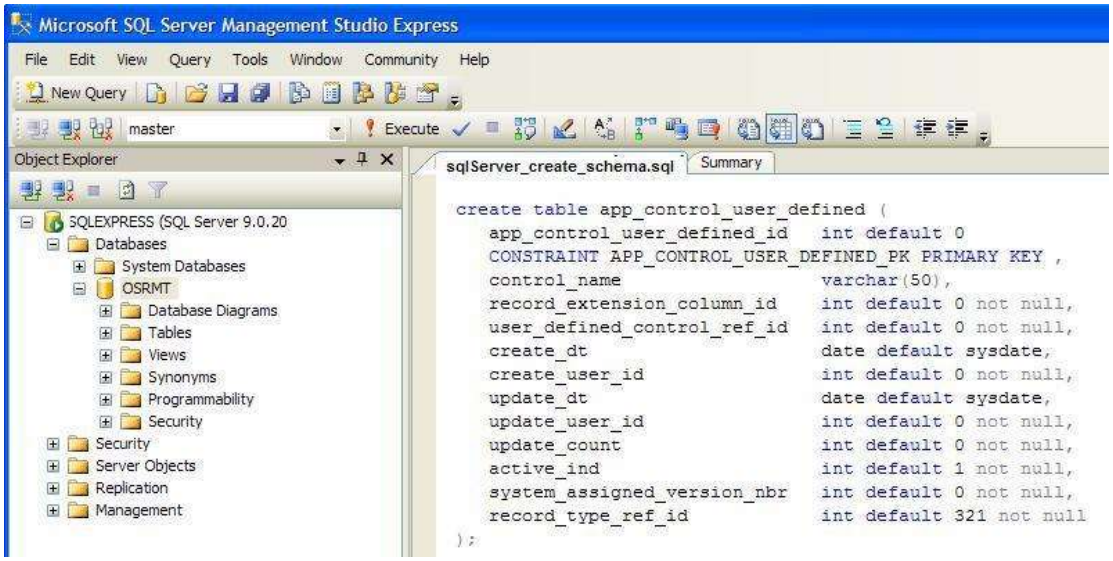
create login osrmt with password = 'osrmt1024'; create user osrmt for login osrmt;

grant select, insert, update, delete to osrmt;

3.4.4 Create Schema

Select the target database from the drop down list – **change from master to OSRMT**

Use menu File-->Open-->File to browse and open the file sqlServer_create_schema.sql



Press the Execute toolbar button to run the script and create all the tables and indexes.

Open the file

sqlServer_create_view.sql

Copy and paste each create statement to create the views manually one by one.

Enable remote access

Select menu Start → Sql Server → Sql Server Surface Configuration

Select Surface Area Configuration for Services and Connections

Select Remote Connections

Select menu **Start → Sql Server → Sql Server Configuration Manager**

Network Configuration →Protocols

Enable TCP/IP

Version 1.8

IP Address blank out Dynamic ports

Set port to 1433 for all the IP Addresses

Restart the SQL Server instance

Change the properties of the SQL Server browser to be started manually

Start the Server Server Browser

Test your connectivity from the command line

sqlcmd -S.\sqlexpress

If you get a prompt

1)

Then, type **exit** and continue.

3.5 PostgreSQL Database Setup Notes

3.5.1 Download and install PostgreSQL 8.1.x or higher

Install PostgreSQL.

3.5.2 Create OSRMT Database

How to create the OSRMT Database on a PostgreSQL 8.1 database server

This illustrates creation of all relevant object in PostgreSQL using the

psql command line client, since this should be available under all circumstances.

However, it can easily done with other tools as well.

Connect to the database server with

a user (login role) which is allowed to create users and databases, e.g. postgres

to any existing database, e.g. template1

using your favourite tool, e.g. the psql command line client, like:

Prompt> psql -U postgres template1

Execute the scripts in the following order

<schema_path>

postgresql_create_user.sql to create the User (Login Role) "osrmt";

in psql: Prompt# \i postgresql_create_user.sql (if you started psql from the same folder where the scripts lie)

postgresql_create_database.sql to create the database "osrmt"; in psql: Prompt# \i postgresql_create_database.sql (see comment for (a)) If you do not want the database to be created in the default tablespace, you have to edit the respective line (currently line 11) in the script.

Connect to the newly created database "osrmt", ideally as user "osrmt"

in psql: Prompt# \c osrmt osrmt -- [First: database, second: user]

4. execute the scripts in the following order

c) postgresql_create_schema.sql to create the tables in schema "public";

in psql: Prompt# \i postgresql_create_schema.sql (if you started psql from the same folder where the scripts lie)

postgresql_create_view.sql;

in psql: Prompt# \i postgresql_create_views.sql (see comment for (a))

3.5.3 To uninstall and remove everything from the database

Connect to another database with a user which is allowed to create users and databases, just like under (1),

with psql: Prompt> psql -U postgres template1

Run the script postgresql_drop_all.sql;

in psql: Prompt# \i postgresql_drop_all.sql

The database with all objects and date and the user (login role) are dropped. Caution: The data are lost, and would have to be restored from a backup, if there is one!

3.6 Import database contents

Attention!	connection.xml file location: Web application - jboss-4.0.3/bin/connection.xml; Desktop application - /connection.xml, that is in the root folder Further will be called just **connection.xml** --- upgrade.bat/sh script location: Web application - jboss-4.0.3/bin/upgrade.bat; Desktop application - /upgrade.bat, that is in the root folder. Further will be called **upgrade.bat/sh** --- **Note**: For Linux/Mac OS users - execute 'chmod +x upgrade.sh' before running upgrading --- Edit the connection.xml in the client directory as described in section **4 (important!)** to point to the new database.

Run upgrade.bat/sh

Select option 4 in the following command line prompt

"Select configuration option 1,2,3 or 4

1) Define a new connection

2) Test the connection

3) Save the new connection

4) Initialize a new database

5) Upgrade 1.3 to 1.4 database

6) Migrate database contents

7) Export language file

8) Import language file

0) Exit

Enter option number [Exit]:"

Confirm the database to be initialized (restart upgrade.bat if you changed connection.xml)

The utility checks and will error if the database tables are not empty.

Enter option number [Exit]: 4

initializing database defined in connection.xml:

jdbc:mysql://localhost/osrmt?useUnicode=true&characterEncoding=UTF-8

osrmt

mysql

Target correct? Y|N [Y]: y

Empty schema located - initialize and populate schema? [Y]:

Pressing Y and enter on the last prompt will update the database and return to the configuration menu.

Instructions to connect the 2 tier application client to the new database

You need the client to connect to the database in order to populate the database with essential reference data.

Edit connection.xml. You may also delete that file and copy one of the existing connection <supporteddb>.xml files to become connection.xml. e.g.

Copy connection.mysql.xml connection.xml

The file has a ***ConnectionProperty*** section which is repeated once per connection in between <object> tags.

The properties for each connection are as follows.

Property	**Description**	**Example(s)**
accessSequence	Future – sequence connections are presented	
active	Only active connections are made available to the client or Server	**True**
connectionType	Method of connecting – Jdbc for a 2 tier client or the server. J2ee for a 3 tier client.	Jdbc
		J2ee
connectToURL	Specifies if the Database Driver will use a class loader with the URL	False
driverClass	Database Driver	oracle.jdbc.driver.OracleDriver
		com.mysql.jdbc.Driver
		com.microsoft.sqlserver.jdbc.SQLServerDriver
		sun.jdbc.odbc.JdbcOdbcDriver
		org.postgresql.Driver
environment	Custom name displayed in the login environment list	Production
		Demo
url	Database connection	jdbc:oracle:thin:@localhost:1521:inst1

	Information	jdbc:mysql://localhost/osrmt
		jdbc:sqlserver://localhost;databaseName=osrmt
		jdbc:odbc:DRIVER={Microsoft Access Driver
		(*.mdb)};DBQ=demo.mdb
		jdbc:postgresql://localhost/postgres
username	Database connection username	orauser
		osrmt
		Osrmt
		Admin
		Osrmt
Password	Database connection password	Password you choose

Note that with version 1.8 the database password is not obscured with 2 tier application client connection.xml

With the 3 tiered setup you need to set the connection type to J2ee and you do not need to fill out the database fields.

Note: With the 3 tiered the connection information is stored on the server and not in "connection.xml"

5. Install the j2ee OSRMT server for a 3 tier configuration

This step is required only for the 3 tier configuration.

5.1 Determine current environment

Validate that Java version 8.0 or higher is installed (see previous instructions). Run command line and execute next command

java -version

you will see Java version currently installed. If Java lower than 1.8 is installed you will need to upgrade it.

5.2 Install Software

Download OSRMT Web application and extract it.

Shortcuts can be created manually point to

<extracted_directory>\jboss-4.0.3\bin\run.bat

Linux addendum

Sudo chmod +X run.sh

5.3 Network Parameters (JNDI)

The default installation of JBoss will configure the server to start listening to Port 1099.

If your OSRMT server is running on a machine different to your client note the IP Address of your server using ipconfig from the command line.

To change the port the JBoss application server listens on edit the file

<extracted_directory>jboss-4.0.3\server\default\deploy\naming-service.xml

Replace 1099 and possibly 1098 with alternate ports. Make a note of the port 1099 is changed to.

Edit the client file to match the port

<desktop_app_extracted_folder>\dd\client\jndi.properties

Ensure that java.naming.provider.url is configured with the right IP and port e.g.

java.naming.provider.url=jnp://10.252.168.247:1099

TIP

If you have an existing Application Server, Jboss or otherwise you can deploy

5.4 Start Server

Prior to starting the server run the client and **ensure connectivity** to the database using the same **connection.xml**

Execute

<web_app_extracted_folder>\jboss-4.0.3\bin\run.bat

Note any errors in the console or written to systemlog_details.txt

Note below instead of executing **run.bat** for desktop app, execute **run3tier.bat**

Linux addendum

Sudo chmod +X run3tier.sh

6. Connect the 3 tier application client to the J2ee server

6.1 Configure connection

Replace the connection.xml with the file connection.3tierJ2ee.xml

Edit <desktop_app_folder>\dd\client\jndi.properties and put an appropriate IP or hostname to replace 127.0.0.1

6.2 Start 3 tiered client

You must execute

run3tier.bat

Linux addendum

Sudo chmod +X run3tier.sh

Which will then have the client connect to the server - barring any ip, network, firewall, or other issues.

Note that if for instance the server fails to connect to the database, the client will log those errors too - which may seem confusing.

7. Use web GUI working with J2ee server

URL

The URL for clients to connect is http://<host>:8080/osrmt

e.g. http://127.0.0.1:8080/osrmt

8. Troubleshooting

8.1 Diagnose client connectivity issues

The default shortcuts point to the 2 tiered version

Execute

<extract_folder_path>\run.bat

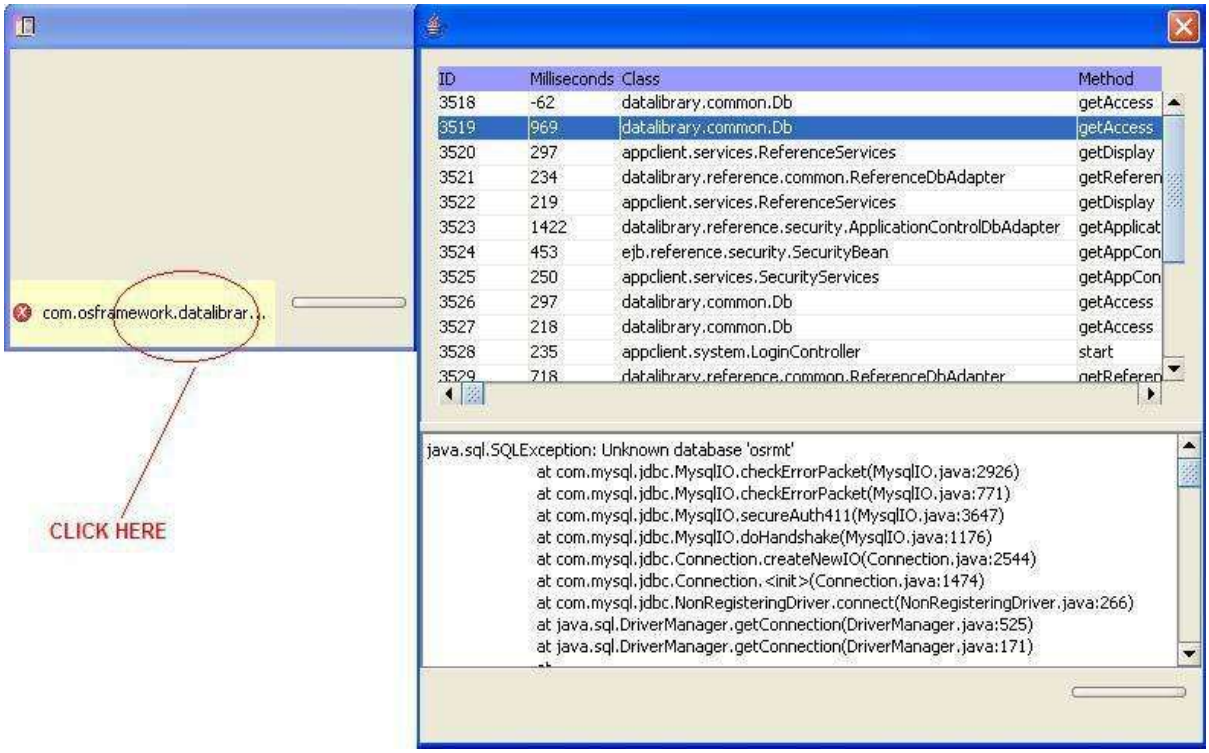
The login controls, labels and default username are read from the first successful active database in connection.xml

If there are connection errors in connection.xml, connectivity issues or errors within the database you can click on the status bar to open the system log.

Alternatively, example the file <**extract_folder_path**>\system_ details.log The file is binary fixed size but you can view with a text editor.

You can delete system.log and system_details.log as they will be recreated when needed.

See next page for System Log user interface.



Click on the list entries to view the details of the error (you often need to scroll to the lower pane to find out what started the stack trace.

The system log retains all errors within the last 5 minutes starting with the earliest. So please make sure that you are on the right entry, which is usually the second **datalibrary.common.DB** entry for database connectivity issues.

8.2 Diagnose server connectivity issues

The server will write database errors to the console – otherwise edit the binary system log files.

Specifically open the fixed size **system_details.log** and scroll down to the last few entries. You can delete both system.log and system_details.log to have the files recreated (don't delete just one…)

Appendix I – MySQL Database Setup Notes

mySQL Sample connection.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<java version="1.5.0_04" class="java.beans.XMLDecoder">
<object class="com.osframework.datalibrary.common.ConnectionProperty">
<void property="accessSequence">
<int>0</int>
</void>
<void property="active">
<boolean>true</boolean>
</void>
<void property="connectionType">
<string>Jdbc</string>
</void>
<void property="connectToURL">
<boolean>false</boolean>
</void>
<void property="driverClass">
<string>com.mysql.jdbc.Driver</string>
</void>
<void property="environment">
<string>mysql</string>
</void>
<void property="url">
<string>jdbc:mysql://localhost/osrmt</string>
</void>
<void property="unicodeConnection">
<boolean>true</boolean>
</void>
<void property="username">
```

<string>osrmt</string>

</void>

<void property="password">

<string>osrmt</string>

</void>

</object>

</java>

Appendix II – Oracle Database Setup Notes

Oracle sample connection.xml

```
<?xml version="1.0" encoding="UTF-8"?>

<java version="1.5.0_04" class="java.beans.XMLDecoder">

<object class="com.osframework.datalibrary.common.ConnectionProperty">
<void property="accessSequence">
<int>0</int>
</void>
<void property="active">
<boolean>true</boolean>
</void>
<void property="connectionType">
<string>Jdbc</string>
</void>
<void property="connectToURL">
<boolean>false</boolean>
</void>
<void property="driverClass">
<string>oracle.jdbc.OracleDriver</string>
</void>
<void property="unicodeConnection">
<boolean>false</boolean>
</void>
<void property="environment">
<string>oracle</string>
</void>
<void property="url">
<string>jdbc:oracle:thin:osrmt/osrmt@192.168.1.101:1521:admin</string>
```

```
</void>
<void property="username">
<string>osrmt</string>
</void>
<void property="password">
<string>osrmt</string>
</void>
</object>
</java>
```

Appendix III – MS SQL Server Database Setup Notes

SqlServer sample connection.xml

```
<?xml version="1.0" encoding="UTF-8"?>

<java version="1.5.0_04" class="java.beans.XMLDecoder">

<object class="com.osframework.datalibrary.common.ConnectionProperty">
<void property="accessSequence">
<int>0</int>
</void>
<void property="active">
<boolean>true</boolean>
</void>
<void property="connectionType">
<string>Jdbc</string>
</void>
<void property="connectToURL">
<boolean>false</boolean>
</void>
<void property="driverClass">
<string>com.microsoft.sqlserver.jdbc.SQLServerDriver</string>
</void>
<void property="environment">
<string>sqlserver</string>
</void>
<void property="url">
<string>jdbc:sqlserver://localhost;databaseName=osrmt</string>
</void>
<void property="unicodeConnection">
<boolean>false</boolean>
```

```
</void>
<void property="username">
<string>osrmt</string>
</void>
<void property="password">
<string>osrmt1024</string>
</void>
</object>
</java>
```

Appendix IV – PostgreSQL Database Setup Notes

PostgreSQL Sample connection.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<java version="1.5.0_04" class="java.beans.XMLDecoder">
<object class="com.osframework.datalibrary.common.ConnectionProperty">
<void property="accessSequence">
<int>0</int>
</void>
<void property="active">
<boolean>true</boolean>
</void>
<void property="connectionType">
<string>Jdbc</string>
</void>
<void property="connectToURL">
<boolean>false</boolean>
</void>
<void property="driverClass">
<string>com.mysql.jdbc.Driver</string>
</void>
<void property="environment">
<string>mysql</string>
</void>
<void property="url">
<string>jdbc:mysql://localhost/osrmt</string>
</void>
<void property="unicodeConnection">
<boolean>true</boolean>
</void>
<void property="username">
```

<string>osrmt</string>

</void>

<void property="password">

<string>osrmt</string>

</void>

</object>

</java>

Appendix V – MySQL and PostgreSQL connectivity

Evaluation using Both , PostgreSQL and Mysql.

The machine has Debian GNU / Linux 3.1 Sarge (up to dated) with Sun Java 1.5.0_08 packaged in .deb with java-package.

The Mysql must have UTF8 support. So, it **must** be 4.1.x or newer.

I installed the mysql 5.0.24a-3 from http://www.backports.org

Also, the OSRMT must work with Postgresql 8.1.x, so installed PostgreSQL 8.1.4_6 from http://www.backports.org .

In Debian GNU / Linux servers, security is spartan.

The Postgresql is preconfigured to only accept local socket connections.

You will have to configure /etc/postgresql/8.1/main/pg_hba.conf and /etc/postgresql/8.1/main/postgresq.conf to

allow tcp/ip connections from your test machines addresses (10.200.27.17, 10.200.27.40 are only examples) and

appropriate authentication method.

/etc/postgresql/8.1/main/postgresql.conf snipet:

Code:

```
#---------------------------------------------------------------------------
# CONNECTIONS AND AUTHENTICATION
#---------------------------------------------------------------------------

# - Connection Settings -

#listen_addresses = 'localhost'          # what IP address(es) to listen on;
comma-separated list of addresses;
defaults to 'localhost', '*' = all listen_addresses = 'localhost,10.200.27.17,10.200.27.40' port = 5432
```

/etc/postgresql/8.1/main/pg_hba.conf snipet:

Code:

```
# TYPE  DATABASE        USER            CIDR-ADDRESS          METHOD

# "local" is for Unix domain socket connections only
#local   all             all                                     ident sameuser
local   all             all                             password
# IPv4 local connections:
host    all             all             127.0.0.1/32            md5
host    all             all             10.200.27.17/32         md5
host    all             all             10.200.27.40/32         md5
```

The mysql server is also locked in local connections.

One of the possible alternatives is to configure the following files:

/etc/mysql/my.cnf

Code:

Instead of skip-networking the default is now to listen only on

localhost which is more compatible and is not less secure.

```
#bind-address           = 127.0.0.1
```

AFM 26set2006 bind address comented out to allow remote connections.

see also hosts.allow and hosts.deny

/etc/hosts.deny

Code:

```
```

AFM 26set2006 control over mysql connections in tandem with hosts.allow mysql:ALL

/etc/hosts.allow

Code:

AFM 26set2006 allow only selected IPs. mysql:10.200.27.17 10.200.27.40 127.0.0.1

These settings are **not** recommended for production servers. They are **only enough for evaluation testing** on a non-production critical spare server.

For production, you will also need additional security measures.

The first run is quirky.

You will NOT see correct dialog boxes and buttons.

The OSRMT will detect an empty schema and will load data.

Then, an "empty" pop up, with 2 tiny buttons will appear.

The right one is the ok (I guess). The program finish.

Restart it and a log in window with DEMO user will appear. Password is "demo".

The process for the other database is similar.

If you use only one database, edit connection.xml and REMOVE the unused database parameters.

Appendix VI- Database Installations

MySQL Installation

Download and install MySQL 5.0 or higher from the following Link

https://dev.mysql.com/downloads/

Install MySQL

Create OSRMT Database

Command Line Instructions below

Start > Program files\MySQL\MySQL Command line client

create database osrmt;

show databases;

Create Schema

From the command line execute the create schema and create view script source <path>

connect osrmt;

>source C:\Program Files\osrmt\v1_50\client\schema\mysql_create_schema.sql >source C:\Program Files\osrmt\v1_50\client\schema\mysql_create_view.sql

Please ignore warning message when dropping view ERROR 1051 (42S02): Unknown table 'osrmt.artifact detail'

Create Login

create user osrmt identified by 'osrmt';

grant all on osrmt.* to osrmt;

Oracle Installation

Download and install Oracle Express Database from the following link

https://www.oracle.com/technetwork/database/enterprise-edition/downloads/index.html

Install Oracle Express and create a database instance.

Create Login

Execute the below instructions from command line

sqlplus system/manager

create user osrmt identified by osrmt

default tablespace <some tablespace your choice> temporary tablespace <some tablespace your choice>; grant dba to osrmt;

Create Schema

From the command line execute the create schema and create view script source <path>

sqlplus osrmt/osrmt

>@oracle_create_schema.sql

>@oracle_create_view.sql

Create database tables and indexes

Execute the shipped <db>_create_schema.sql script which can be found in the schema directory e.g.

<client directory>\v1_50\client\schema\oracle_create_schema.sql

<client directory>/osrmt/client/schema/mysql_create_schema.sql etc.

Create database views

Execute the shipped <db>_create_view.sql script which can be found in the client schema directory e.g.

<client directory>\v1_50\client\schema\sqlserver_create_view.sql etc.

TIP

Ignore any warnings that states ***"the view did not previously exist"***