

FUSION REGISTRY

Using a SQL Server database with Fusion Registry

FUSION REGISTRY
VERSION 10

SQL Server Deployment Notes

This document provides guidance on installing and configuring Fusion Registry with SQL Server

Contents

1	Overview	3
2	SQL Server JDBC Driver	4
3	Configuring the SQL Server Connection.....	5
4	SQL Server Connection String	6
5	Configuring for Single Sign-On (SSO)	7
6	Troubleshooting.....	9
6.1	Unable to communicate with SQL Server	9
6.2	No DLL specified for Authentication	9

Version History

Version #	Implemented By	Revision Date	Reason

1 Overview

Fusion Registry supports SQL Server for its main operational database, in addition to MySQL and Oracle.

The operational database is used for:

- Storage of the SDMX Structural Metadata
- Persistence of observation data that has been loaded into the 'Fusion Data Store' in-memory data
- Software configuration settings
- Audit information

2 SQL Server JDBC Driver

TODO....

Driver Requirements

Driver is supplied with Fusion Registry. At the current time the driver is XXXX. If you need to upgrade this driver then use....

Fusion Registry supports the 'ojdbc8' Thin JDBC Driver with Java JRE / JDK versions 1.8 or 1.10.

Note that JRE 1.9 is not supported.

The driver consists of a single 'jar' file: 'ojdbc8.jar'.

The 'jar' file is not included as part of the Fusion Registry distribution so must be obtained and installed separately prior to starting the application.

'ojdbc8.jar' can be downloaded from Oracle at:

<https://www.oracle.com/technetwork/database/features/jdbc/jdbc-ucp-122-3110062.html>

Driver Installation

The 'ojdbc8.jar' must be placed in the Java CLASSPATH.

The recommended option is to add the filesystem location of the 'ojdbc8.jar' to the CLASSPATH of the application server. If using Apache Tomcat, this can be done by setting the CLASSPATH environment in the Tomcat ~/bin/setenv.sh script.

For example:

```
CLASSPATH=/home/oracle/jdbc/ojdbc8.jar
```

..where /home/oracle/jdbc is the directory containing the 'ojdbc8.jar' file.

An alternative option is to copy 'ojdbc8.jar' into Fusion Registry's deployment FusionRegistry/WEB-INF/lib directory. If using Tomcat, this can be found in the webapps folder. For instance:

```
/home/tomcat/webapps/FusionRegistry/WEB-INF/lib
```

This approach is not recommended because the 'ojdbc8.jar' file will be deleted every time a new version of the Fusion Registry web application is deployed.

3 Configuring the SQL Server Connection

Fusion Registry has a user interface to specify the credentials for the SQL Server database

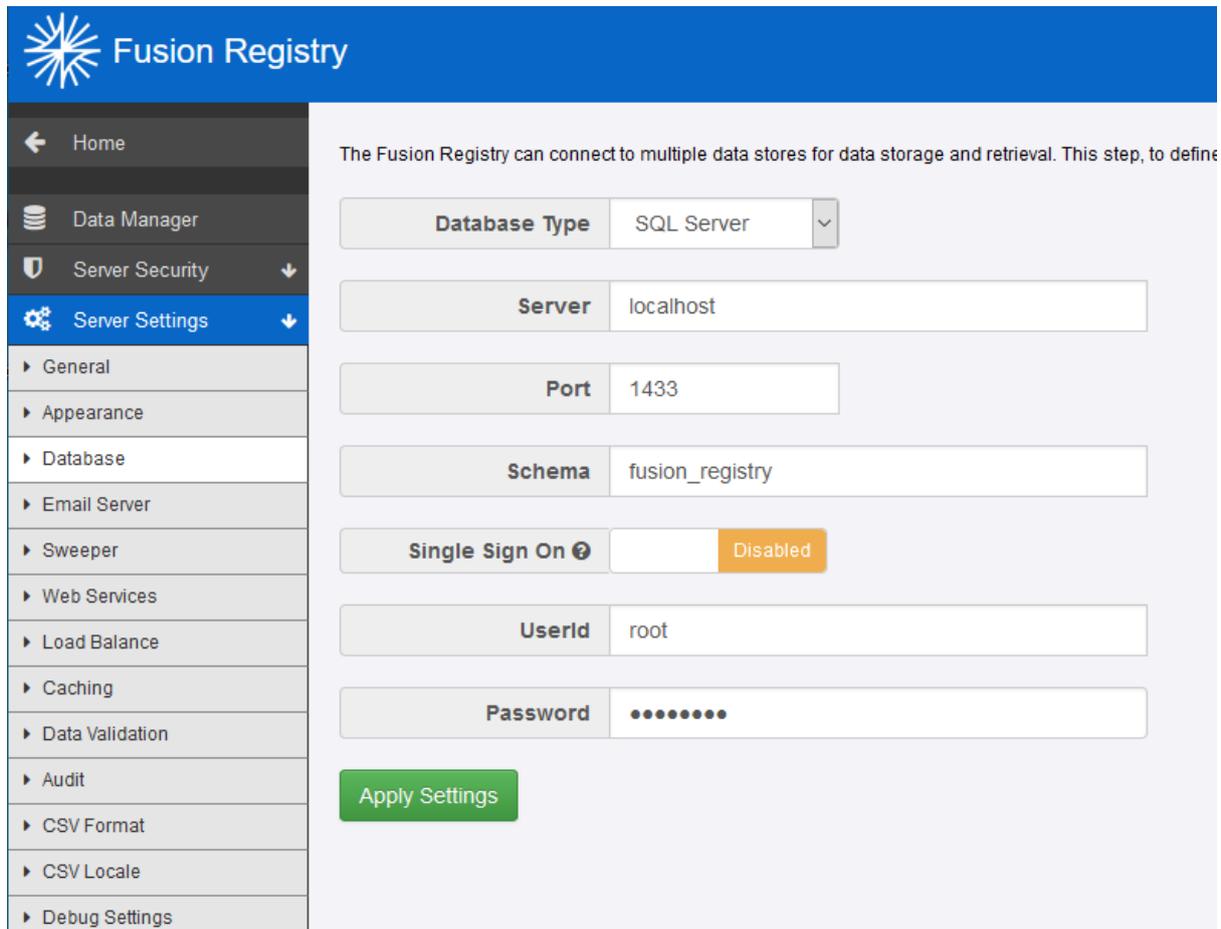


Figure 1 - showing Fusion Registry database configuration

Database Type	Select 'SQL Server'
Server	Supply a resolvable hostname or IP address for the SQL Server database server
Port	The IP port of the database service – 1433 is the default
Schema	The Database within SQL Server to connect to
UserId	The id of a pre-existing SQL Server user with sufficient privileges to create and delete tables, and insert, update and delete data
Password	The password for the named SQL Server user

4 SQL Server Connection String

Fusion Registry uses the database configuration information supplied at install to construct a JDBC connection string which is stored in a configuration properties file.

If problems are experienced attempting to establish a connection to the SQL Server database service, administrators are advised, as part of the fault-finding process, to examine the connection string and verify that it is as expected.

Location of the Properties File Containing the Connection String

From release 9.3, the Fusion Registry properties file is, by default, placed under the MetadataTechnology directory in the home directory of the user that owns the web application server.

For example, in an Apache Tomcat installation, the path to the properties file could be:

```
/home/tomcat/MetadataTechnology/FusionRegistry/fusion_registry.properties
```

where /home/tomcat is the home directory of the 'tomcat' user which owns the application server.

What to Look for in the Properties File

The properties relating to the SQL Server database should look similar to the following example:

```
database.url=jdbc\:sqlserver\://localhost\:1433;databaseName\=fusion_registry;integratedSecurity\=true;  
database.password=8VpAqQJWqZI\=  
database.username=fusion_registry
```

database.url	The database.url property holds the SQL Server connection string and should be structured to the SQL Server specification. Colons and other characters with special meanings in the URL syntax are escaped using a backslash.
database.password	Note that the database user password is encrypted using a one-way hash so it cannot be changed by editing the properties file.

5 Configuring for Single Sign-On (SSO)

Fusion Registry supports Single Sign-On (SSO) when connecting to a SQL Server database. This can be activated via the Registry UI. In order for this feature to work a DLL is also required.

The Fusion Registry can connect to multiple data stores for data storage and retrieval. This step, to define a database connection, is for the storage of the Registry structures.

Database Type	SQL Server
Server	localhost
Port	64771
Schema	my-schema
Use Single Sign On	Using SSO
Userid	
Password	

Apply Settings

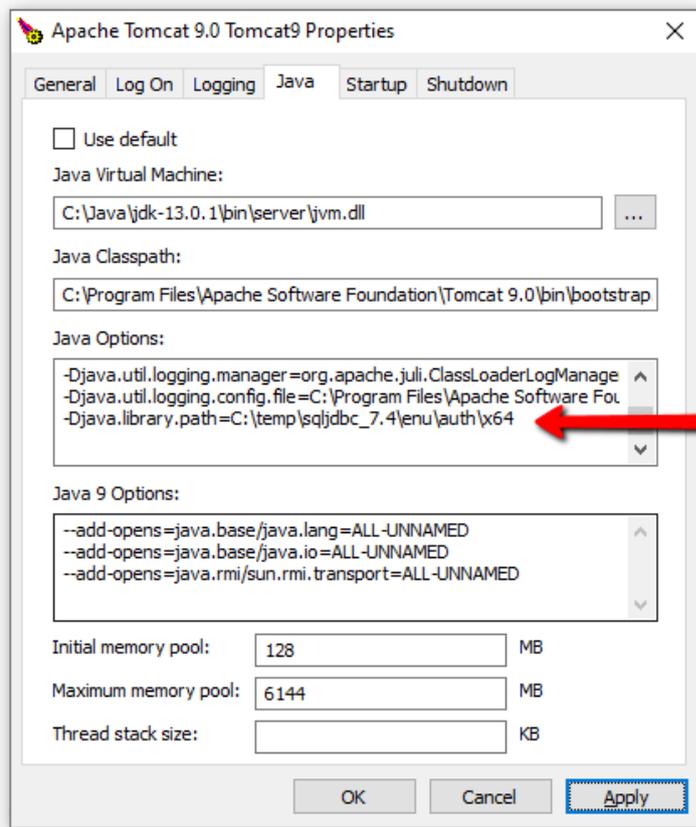
Figure 2 showing the database settings page for a SQL Server database type

The DLL can be obtained from Microsoft. You will need to download the "Microsoft SQL Server JDBC Drivers" package which contains a number of drivers named "sqljdbc_auth .dll" but for different systems (e.g. x86, 64 bit, etc.). You need to locate the appropriate DLL for your system.

This DLL needs to be supplied to the Java Runtime running your Web Application Server. There are a number of ways in which this can be achieved. Two of the simplest methods are listed below:

1. Copy the DLL file to the Java Runtime "bin" directory that is running your Web Application Server. It is important to place the DLL in the correct directory (for example: C:\Java\jdk1.8.0_92\jre\bin). Note: that modifying a Java Runtime in this manner means that all applications that use this Java Runtime will be affected.
2. Pass the DLL location to the Web Application Server on server startup. Locate the directory with the DLL you wish to add (e.g. c:\temp) then add the following line to setenv.bat and the Java library path will be modified allowing Tomcat to access the DLL file:
`set CATALINA_OPTS=%CATALINA_OPTS% -Djava.library.path=C:\temp\SSO_DLL`

If you are running Apache Tomcat as a service, do not modify setenv.bat. Instead you will need to run the modify the Apache Configuration via the tool supplied by Apache. In the Java Options area specify the location of the directory.



`java.library.path` specifies location of the DLL

Figure 3 - showing ...

Once your Web Application Server has started and can access the correct DLL, SSO can be enabled via the database settings. When attempting to enable SSO, if you receive an error like the following, then the DLL could not be located or is the wrong version for your system:

```
java.lang.UnsatisfiedLinkError: no sqljdbc_auth in java.library.path
```

6 Troubleshooting

6.1 Unable to communicate with SQL Server

The following shows a typical error when the Fusion Registry cannot communicate with the SQL Server database due to invalid credentials or incorrect database name. Note in this image the connection has been attempted to the database "fusion_registry"

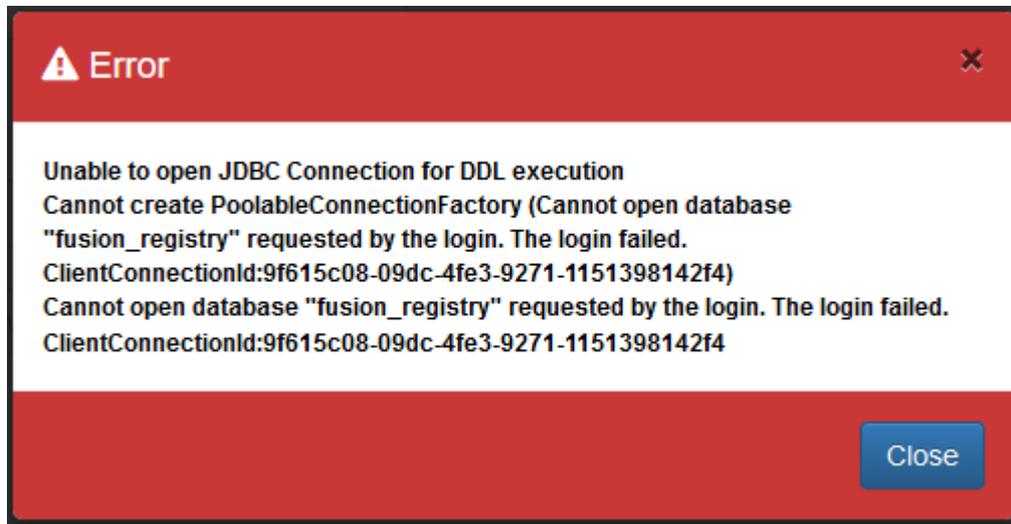


Figure 4 - showing a failed connection due to credentials

To solve this issue check that the supplied credentials are correct and that your SQL Server database can resolve these issues.

If you are using SSO, check that the correct user has been specified..... TODO running as a service

6.2 No DLL specified for Authentication

The following error dialog shows the situation where Single Sign-On has been attempted

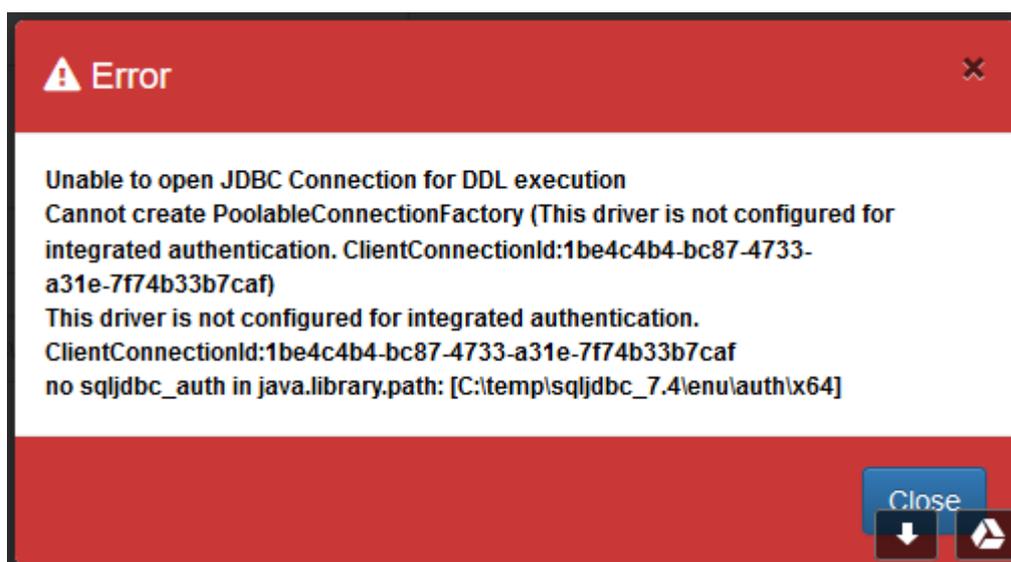


Figure 5 – showing

TODO.....