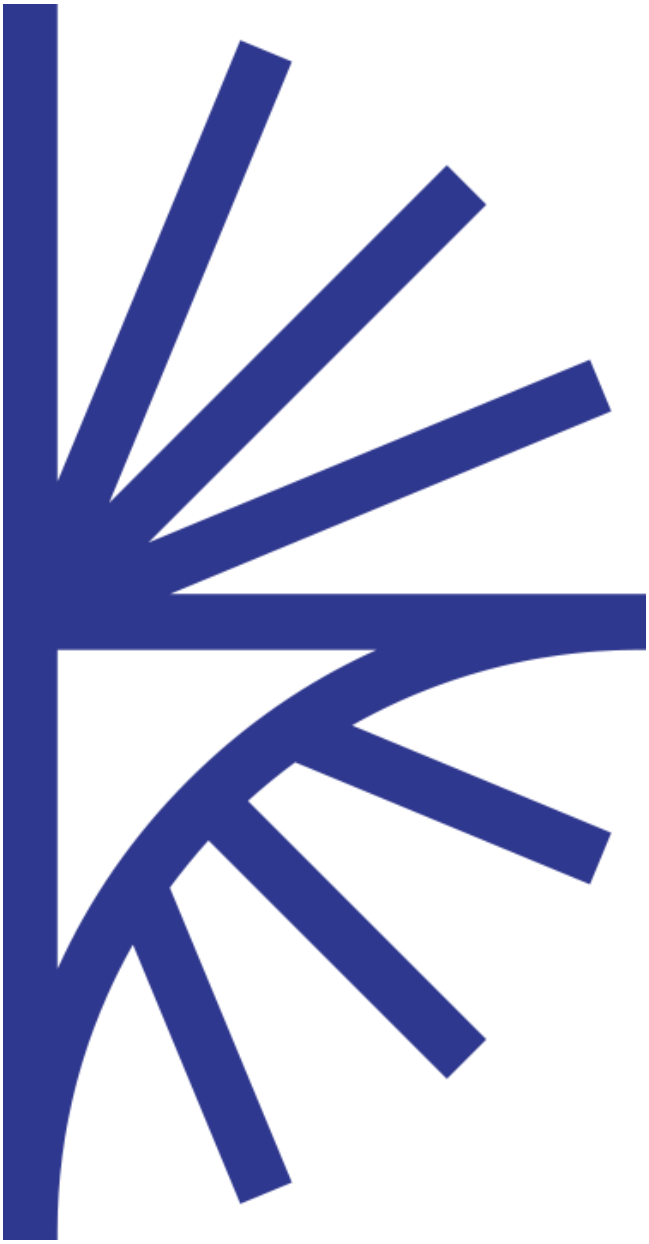

FUSION REGISTRY COMMUNITY EDITION SETUP GUIDE

FUSION REGISTRY COMMUNITY EDITION
VERSION 9

Setup Guide

This guide explains how to install and configure the Fusion Registry.

Fusion Registry: 9.8.x
Document Last Updated: October 22nd 2019

Version: 20191022

Contents

Version History

Version #	Implemented By	Revision Date	Reason
20191022	Phil Lazarou	22nd Oct 2019	Updated Install Section and Security Section

1 Introduction

1.1 Scope of this document

The purpose of this document is to provide an overview of how to install Fusion Registry in a Java Servlet Container. This document explains the Fusion Registry install process, how to configure the Fusion Registry and specific advice on configuring aspects of the Java Servlet Container, Apache Tomcat.

It is recommended that readers have familiarity with the appropriate required dependencies which are listed in the following sections.

1.2 Fusion Registry Distribution

Fusion Registry is distributed as a single Web Application Archive (war) file: *FusionRegistry.war* it should be deployed to a servlet container such as Apache Tomcat (a free product).

1.3 Java (required)

A Java Runtime Environment (JRE) of version 1.8 or higher is required.

1.4 Servlet Container (required)

Fusion Registry is deployed to a Servlet Container. Apache Tomcat is a popular, open source servlet container, download links and installation instructions can be found at the following URL.

http://tomcat.apache.org/

It is recommended to use the latest version of Apache Tomcat as it will include the latest security patches.

Apache Tomcat 8 or above is supported.

The rest of this document will only give servlet container information regarding Apache Tomcat.

1.5 Database (optional)

Fusion Registry makes use of an Object Relational Mapping (ORM) library called Hibernate. This allows Fusion Registry to communicate with any SQL-92 compliant database. This distribution has only been set up to connect to MySQL, Oracle, and SQL Server databases, the minimum tested version for these databases are: MySQL 5.5, Oracle 10g and SQL Server 2010.

If your database is not one of these types, please contact Metadata Technology, as we may be able to add your database to the list of supported database management systems.

1.6 Fusion Security (optional)

By default, Fusion Registry provides the ability to secure the Registry only allowing a single trusted user to perform changes. If you wish to have a number of user accounts with different access credentials, then Fusion Security is required to create and administer users. Please refer to the Fusion Security Setup and User guides as well as the section on Security in this document (see section 3.4).

1.7 LDAP Active Directory (optional)

As an alternative Security mechanism, it is possible to use an LDAP Active Directory to manage user authentication with Fusion Registry. Please refer to the section on Security in this document (see section 8).

2 Deployment

2.1 Choice of Java Servlet Container

Fusion Registry must be run within a Java Servlet Container. Metadata Technology recommends using Apache Tomcat as the Java Servlet Container, as this has been used during the testing lifecycle of Fusion Registry. The Fusion Registry has only been tested in Apache Tomcat and therefore we cannot guarantee that the Fusion Registry will work with other Java Servlet Containers.

2.2 Deployment Using Tomcat

Fusion Registry consists of a single .war file called *FusionRegistry.war*. This file needs to be copied into the directory: *<TOMCAT_HOME>/webapps* then the Tomcat server should be started. As the Tomcat application server starts, the contents of the Fusion Registry war file will be unpacked into the directory:

<TOMCAT_HOME>/webapps/FusionRegistry

Please check the Tomcat log files to ensure that Fusion Registry has deployed correctly. Once it has then you may navigate to the URL:

http://[server]:[port]/FusionRegistry/

The values for server and port must be replaced with the IP address and port number that the web application server is running on. For example, if the web browser is running on the same machine as the web application server and the Apache Tomcat has not had its default port settings modified, then the following address can be used:

http://localhost:8080/FusionRegistry/

2.3 Configuring Tomcat Memory

It is important to override the default Tomcat server memory settings as the default Tomcat settings will not be adequate to run the Fusion Registry. Unless you are running Tomcat as a Windows service (see section 11) overriding the memory settings can be achieved by placing a *setenv.bat* (Windows) or *setenv.sh* (Unix) file into the Apache Tomcat *bin* folder. The recommended minimum settings are:

- **2Gb Heap Memory**

The Fusion Registry distribution contains a *setenv.bat* (Windows) and *setenv.sh* (Unix) with the recommended minimum settings configured. These files can be copied (and optionally modified) to the Tomcat/bin folder before starting the Tomcat instance.

2.4 Configuring Tomcat HTTPS

An HTTPS connection provides a secure connection to the Fusion Registry server, by using the Secure Sockets Layer (SSL) protocol. It is strongly recommended to enforce a HTTPS connection to ensure username and password details are encrypted between client and server. HTTPS connections can terminate at a load balancer, in which case Tomcat can remain on a standard HTTP connection.

Please refer to the Apache Tomcat guide if configuring HTTPS.

3 Installing Fusion Registry

3.1 Install Wizard

Once the Apache Tomcat instance has been configured and started, the Fusion Registry will be accessible via a web browser. On first use, the Fusion Registry will show the Install Wizard, each step of the wizard must be completed in order to configure the Fusion Registry. Once the wizard is complete, the Fusion Registry home page will load, and the install wizard will no longer be accessible. Future modifications to the Fusion Registry configuration can be performed by logging in, and visiting the Settings pages. All the configuration options available in the Install Wizard are accessible via the Settings pages in the Fusion Registry.

3.2 Step 1 – Database Connection

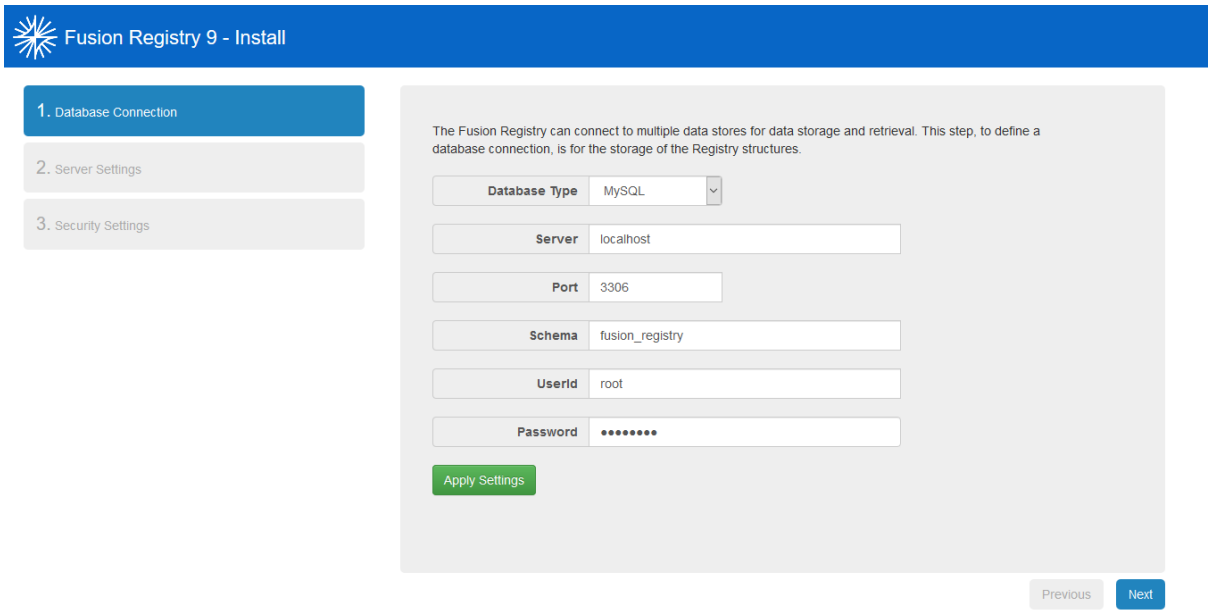

Figure 1 showing step 1 of the install process: Database Connection

The first step of the Install Wizard is to configure a database connection. For relational databases (If (MySQL, SQL Server, Oracle) the Fusion Registry will automatically create the required database tables on connection if they do not already exist.

The in-memory database is not a persistent storage mechanism. Registry settings information is stored in a properties file (discussed in section 5) and as such these settings will be loaded back into memory on tomcat start-up. Any SDMX Structural content loaded into the Fusion Registry is stored in memory and not persisted to any external store, and as such this information will not be preserved on tomcat shut down.

In the database type drop-down there is a choice for “Custom”. This allows the specification of a database connection via a custom string (see section **Error! Reference source not found.** for further d etails on this feature).

3.3 Step 2 – Server Settings

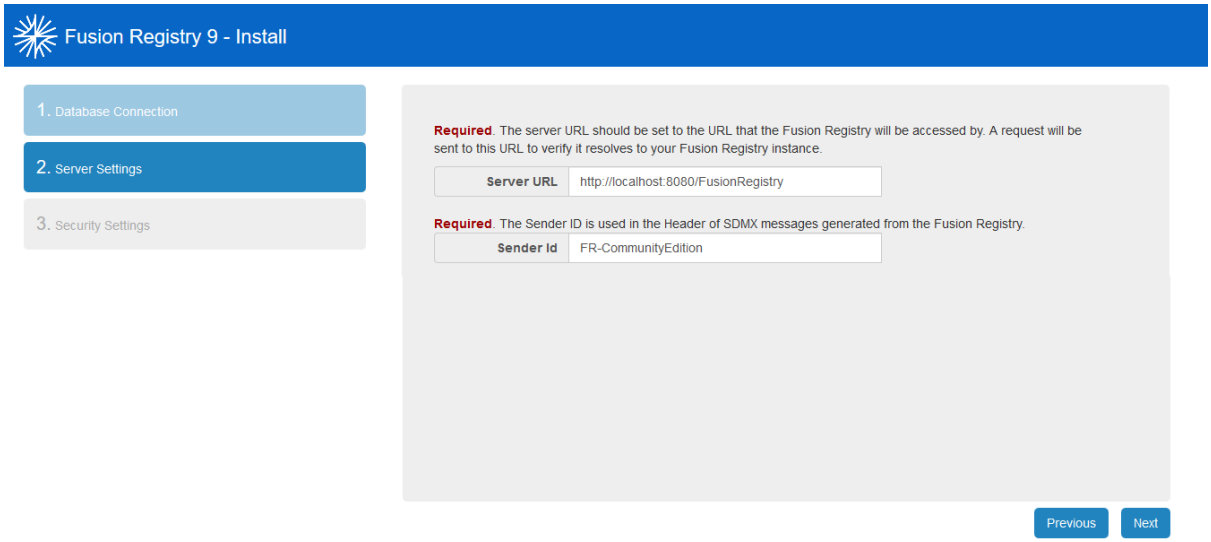

Figure 2 showing step 2 of the install process: Server Settings

The second step allows the user to define two properties:

1. **Server URL** – This is the fully-qualified URL which the Fusion Registry will be access by, and is used by the Fusion Registry when it needs to communicate the URL to a user.

   NOTE: It is very important that the Server URL is set to reflect the public URL that the application will be accessed by, as this will be used as the base URL for any redirects (such as login, logout).

2. **Sender Id** – This Id will be sent in the 'SenderId Header' field of any SDMX messages. The default value is Unknown.

On clicking the *Next* button, the Fusion Registry will validate the Server URL to ensure it is a legal value.

After a successful installation of Fusion Registry, when this page is viewed via the Settings page "General Settings", a third option is displayed on this page: **Default Agency** – When authorized users are creating new structures, the value in this Agency drop-down will be initially set to the value specified here. This drop-down will show all agencies stored in the Fusion Registry, on first install the only Agency contained in this list will be SDMX.

3.4 Step 3 – Security Settings

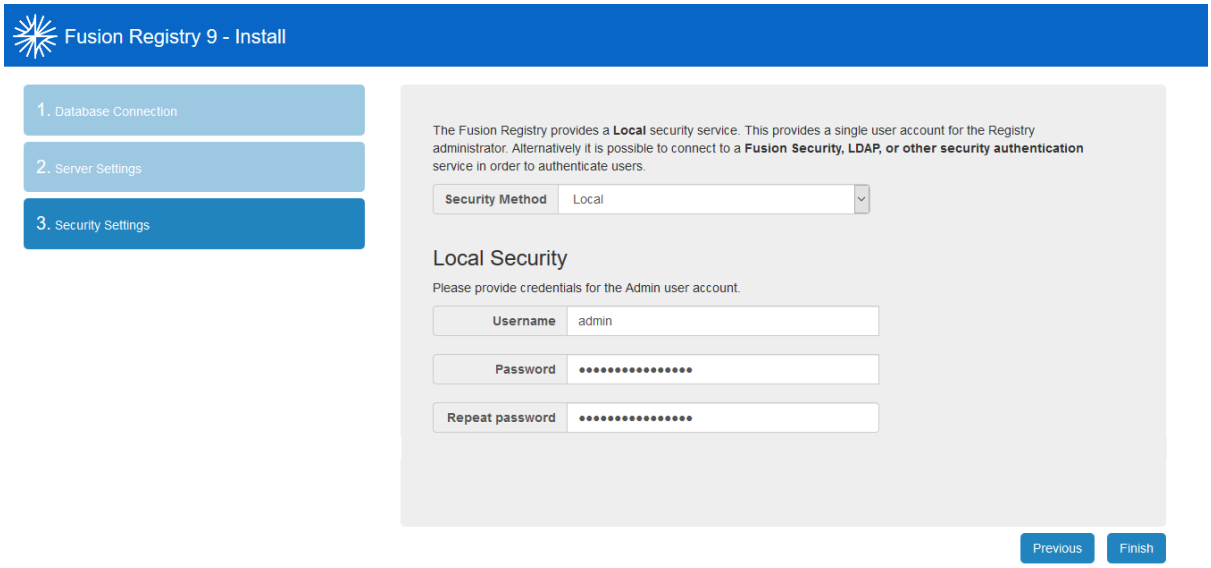

Figure 3 showing step 3 of the install process: Security Settings

The Fusion Registry supports a number of security mechanisms (see later on in this document). On the install pages, the "local root" user credentials need to be specified. This is a single local account that can be used to access the Registry at the highest privilege level of "root". After the install process, only the root user can modify the root user settings, so it is important that the credentials for this account are kept safe.

This step has settings for the root username, password and the maximum number of attempts that can be made to login as root before the account is locked. If this value is set to a zero or negative number, this states there the account will never become locked from repeated failures to login.

If the root user becomes locked, see section 0 for instruction on how to unlock the root user.

4 Installation Completion

On the final step, the *Finish* button completes the Fusion Registry configuration. The browser will redirect the user to the Fusion Registry home page. Fusion Registry configuration settings can be performed by logging in as a user with Admin privileges, and using the Settings menu located on the left hand menu bar.

Note: The settings menu is not visible for non-Admin users

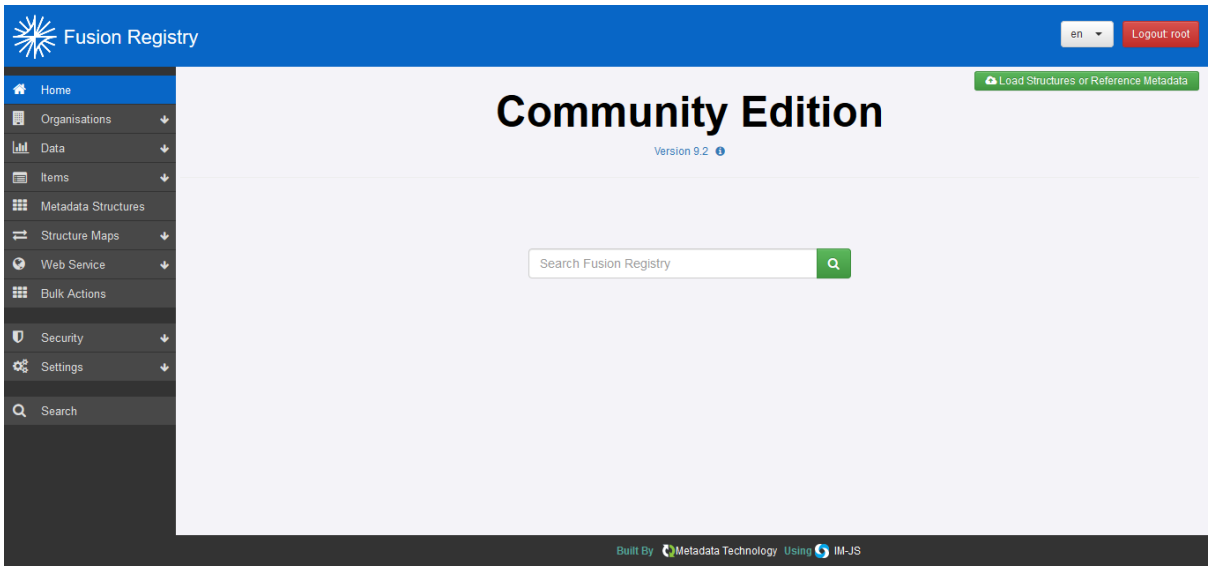

Figure 4 showing the front page of an empty Registry

The front page of the Registry shows the Registry version and a search input field, allowing the user to search for any registered structural metadata.

5 Fusion Registry Properties File

5.1 Introduction

Whilst all of the installation is performed via the Web User Interface, database connection details are stored in a local properties file. If running Fusion Registry with an in-memory database, other settings are also stored in the same properties file.

Unlike previous versions of the Fusion Registry is it recommended that you do not modify the values in this file, as all settings are configured via the web User Interface.

The properties file is only read at start-up and changing any of the values in the Fusion Registry properties file whilst Fusion Registry is running will have no effect.

The properties file is called:

fusion_registry.properties

By default, the Fusion Registry has a copy of this file located in the directory:

<Tomcat HOME>\webapps\<Web AppName>\WEB-INF\classes

If you make any changes using the maintenance tool, Fusion Registry will attempt to save a new properties file to the directory:

<user home>\MetadataTechnology\FusionRegistry

Therefore, on a Windows 7 Operating System this will typically be:

C:\users\<your user name>\MetadataTechnology\FusionRegistry

Whereas on a Unix Operating System, it is more likely to be located at:

/users/<your user name>/MetadataTechnology/FusionRegistry

On Fusion Registry start-up, the Fusion Registry will load the properties file from the WEB-INF\classes directory first, and then look for a properties file in your home directory. If it locates a properties file in your home directory, the values in this file will be read and will override values from the default properties file.

If you are unsure about which of the files Fusion Registry is using to obtain system information, please look at the start-up log in your web application server. There will be entries like the following:

```
INFO localhost-startStop-1
org.springframework.beans.factory.config.PropertyPlaceholderConfigurer -
Loading properties file from class path resource
[fusion_registry.properties]
INFO localhost-startStop-1
org.springframework.beans.factory.config.PropertyPlaceholderConfigurer -
Loading properties file from URL
[file:/C:/Users/<username>/FusionRegistry/fusion_registry.properties]
```

5.2 Changing the location of the properties file

The location of your properties file can be changed. This is useful if you either do not want the Registry to store information in the computer's home directory or if you wish to run multiple Registries on one server.

To specify a new location you need to set a Java System variable called "RegistryProperties" to the URI value of the location where you wish the properties file to be. If you are running Apache Tomcat as a service please refer to Section 9 of this document. Otherwise the easiest way to achieve this is to create a new file called `setenv.bat` (or `setenv.sh` on Unix environments) and place it in the tomcats `bin` directory. The contents of this file should state the full location of the properties file which must be in the URI format. To illustrate this:

```
SET JAVA_OPTS=-DRegistryProperties=file:///c:/dir/AFile.txt
```

(For Windows systems)

```
export JAVA_OPTS=-DRegistryProperties=file:///dir/AFile.txt
```

(For Unix systems)

It is important to note that Fusion Registry will NOT start if this value is incorrect or if this file cannot be created.

To check that this value is being used by the system, check the Registry log during Registry startup and look for a line similar to the following:

```
Property RegistryProperties has been specified as file:///c:/dir/AFile.txt
```

5.3 Overriding Server URL Property

When deploying multiple load-balanced Fusion Registry servers, there may be a requirement for each server to maintain its own value for the "Server URL" property, instead of each accessing a single value from the shared database.

In order to override the Server URL property, edit the Fusion Registry properties file, local to each instance, and modify the file to include a value for 'registry.url'. For example:

```
registry.url=https://localhost:8443/FusionRegistry
```

This property will now be used instead of the database property. In addition, if using Fusion Security, the Fusion Security server domain will be ignored when authenticating a user.

6 Security Roles and Fusion Registry Access

6.1 Overview

The Fusion Registry has 5 distinct user roles, these are:

1. Root
2. Admin
3. Agency
4. Data Provider
5. Data Consumer

Users with Root and Admin roles are able to access all parts of the Fusion Registry application. Only the Root user is able to change the Root user credentials.

A user has a role of **Agency** if their user account is linked to one or more Fusion Registry Agencies. An Agency user has the following privileges:

- Create, modify, and delete SDMX Structures for their own Agency.
- Author and upload Reference Metadata.

Users with roles of Data Provider or Data Consumer have no special privileges when using Fusion Registry Community Edition.

6.2 Registry Security

The Fusion Registry provides the following security mechanisms for authentication:

1. **Local Security** – Local security authenticates the user within the Fusion Registry application. Only one root user account exists and this user has unrestricted access to the whole application. The local root credentials are encrypted and stored in the Fusion Registry database. Local Security is always active but the account can be locked by repeated submissions of incorrect passwords.

One of the following other mechanisms may also be specified:

2. **Fusion Security** –Fusion Security is used to manage user accounts and authenticate users. The Fusion Registry connects to the Fusion Security web application for user authentication. Fusion Security connects provides the ability to set up and maintain user accounts, and link the user to any number of Fusion Registry organisations.
3. **LDAP Active Directory** – Active Directory (AD) is used to connect to a Microsoft Active directory service or equivalent. This allows the registry to both authenticate and authorise users when they log into the registry.

7 Fusion Security

Fusion Security can be specified as an authentication service for Fusion Registry, allowing the management of users and permissions to be controlled from the Fusion Security application.

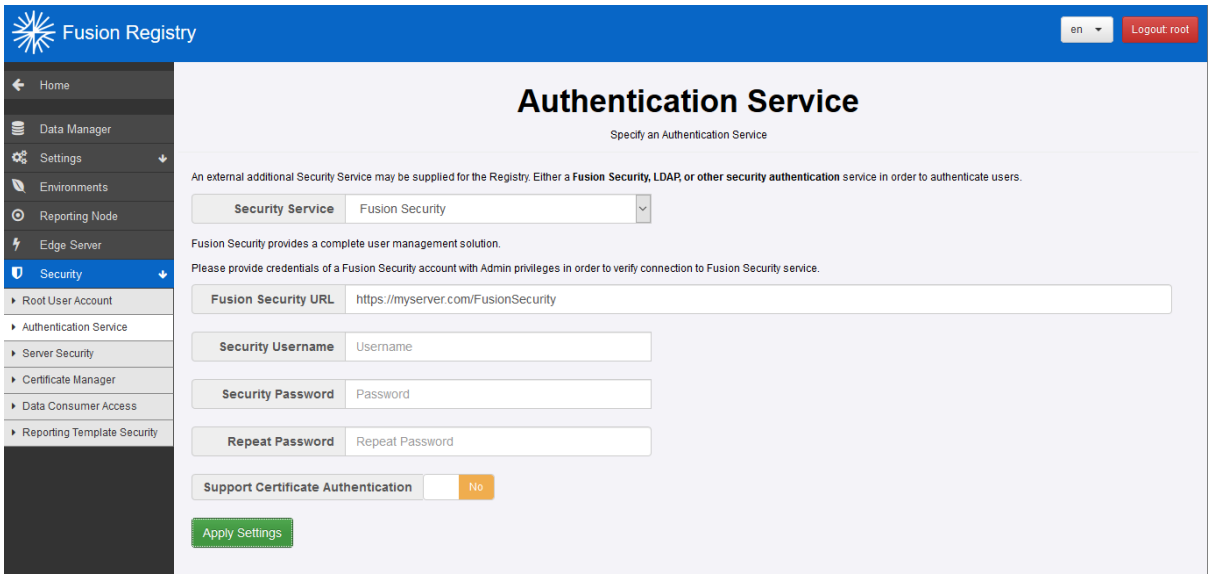

Figure 5 showing Fusion Security being set as an Authentication Service

Navigate to the Authentication Service page via Admin -> Security -> Authentication Service. This page has a drop-down allowing the choice of Security Service to be "none" (no additional authentication service beyond the local root user), "Fusion Security" and "LDAP Active Directory".

Choosing "Fusion Security" as the additional security service requires that you specify the URL of the Fusion Security Service as well as the root credentials to that Security service (note that these are not the root credentials of the local root user of the Fusion Registry). Once "Apply Settings" is clicked, these credentials are checked and if correct then Fusion Security will be configured to be providing security to Fusion Registry.

8 Active Directory

8.1 Overview

Active Directory may be used as the Authentication and Authorisation provider for Fusion Registry. This requires an Active Directory server running on a network that Fusion Registry can communicate to. Fusion Registry can communicate with Active Directory either using a Role Template or else a Role Mapping can be defined.

This document does not explain how to deploy Active Directory or create the required users or groups within it.

8.2 Connection Settings

To establish the connection between Fusion Registry and the Active Directory server, the following information needs to be supplied:

1. Protocol – either LDAP or LDAPS.
2. URL of Active Directory – either the IP address or server name of the Active Directory server.
3. Active directory domain –the domain that Active Directory resides on. This can also be the NetBIOS.
4. Base group DN (optional) - this is the root Distinguished Name (DN) for the Registry to search for groups under. If this is left blank then the search will be performed on the highest level of the Active Directory forest.

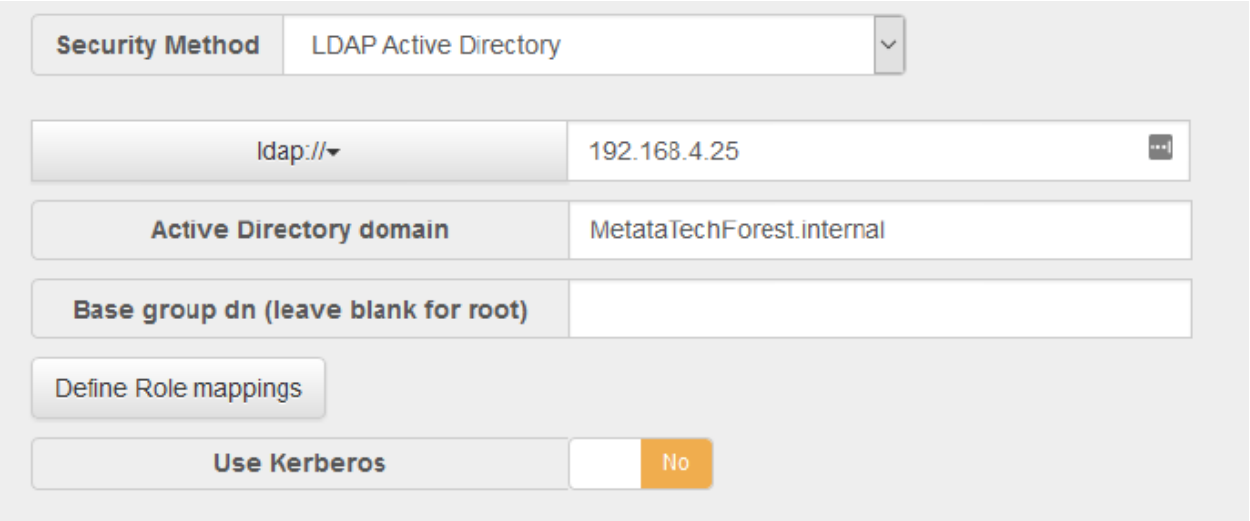

Figure 6 showing the LDAP settings

Important Note: if using the LDAPS protocol then the Registry will be unable to validate this connection fully. This means that if the certificate is not valid for the connection, but the URL is correct, the connection is still considered valid. When you attempt to login to the Registry, it will not work, since the certificate is invalid. It is vital that you ensure you have a valid certificate when using LDAPS.

8.3 Role Template

If no Role Mappings are defined then the Registry will communicate to Active Directory using the Role Template. In this scenario, users are given permission by assigning them to groups and the names of the groups follows a specific pattern to provide authorisation.

To set up roles permitting Agency level authorisation, groups for the appropriate agency must be named “ACY_” and then be followed by the agency name. E.g. A group which permits users assigned to that group to modify SDMX structures, must be named “ACY_SDMX”

To assign a Data Provider role to a user, then the group must be named “DP_” followed by the AgencyId of the Data Provider, an underscore, and then the Data Provider ID. For example to create a group of Data Providers for the Data Provider “DP1” owned by the Agency “ACY”, the group must be named: “DP_ACY_DP1”

To assign a Data Consumer role to a user, then the group must be named “DP_” followed by the AgencyId of the Data Provider, an underscore, and then the Data Provider ID. For example to create a group of Data Providers for the Data Provider “DP1” owned by the Agency “ACY”, the group must be named: “DP_ACY_DP1”.

To create a group that permits Administrator access to the Registry, the group needs to be named “Administrators” (this is not case sensitive).

8.4 Role mapping

If you do not want to map roles to users using the default naming template, you can define your own mapping values. Custom Role Mapping allows an association between a group in Active Directory to permissions in the Registry. For example, you can specify that any users assigned to the group named “ABC” have permission to be Agencies for “ACY1”.

To setup a mapping, click the “Define Role Mappings” button and you will be presented with a dialog titled: “Define Role Mappings”. This modal has controls allowing the creation of custom mappings.

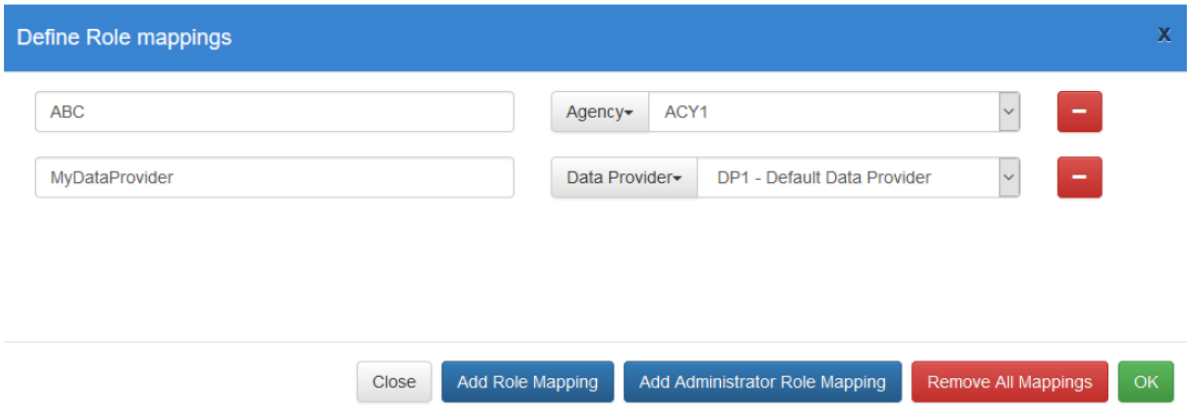

Figure 7 showing the LDAP role mappings

Clicking “Add Role Mapping” presents the user with an input field, and a dropdown:

Figure 8 showing the input field with dropdown

The key is what will be looked up when parsing your roles on AD, the key needs to match the name of the group in Active Directory. The second part is the type of role this mapping value equates to. If for example you want to map “ABC” with the Agency “ACY1”, then you would type “ABC” as the key, and select agency “ACY1” as the value.

When you have added all your required mapping values, clicking “OK” will inform the Registry to use your mapping values and not use the template. Any groups that your Active Directory is connected to and not defined in the mapping will be ignored.

If you wish to define a custom name for an Administrator mapping, click the “Add Administrator Role Mapping”. This will only allow you to specify a key since an admin does not associate with organisations.

9 Single Sign-On (SQL Server only)

Fusion Registry 9 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.

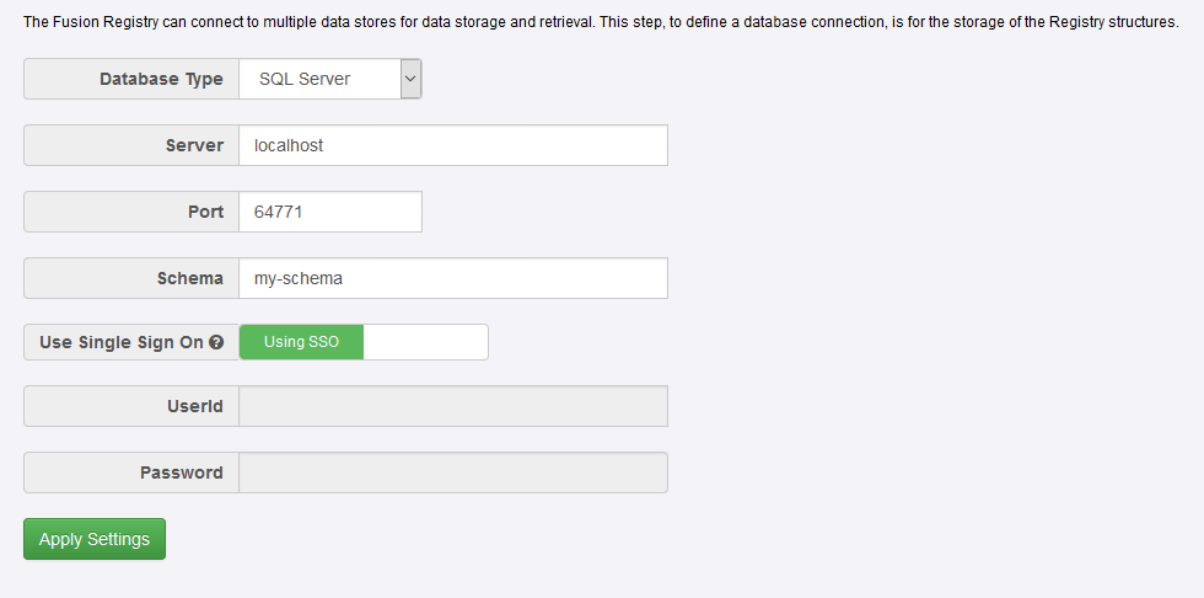

Figure 9 showing the database settings page for a SQL Server database type

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. If you are running Apache Tomcat as a service, please refer to section 11 of this document. Otherwise this can be achieved by modifying the “setenv.bat” file located in the Tomcat bin directory. 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
```

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
```

In this scenario, please double-check the actions you performed and ensure that you are using the correct driver.

10 Additional Settings

10.1 Registry Settings

10.1.1 Accessing Registry Settings

Settings modification can be performed by any Admin user through the links under the Settings tab in the left-hand menu.

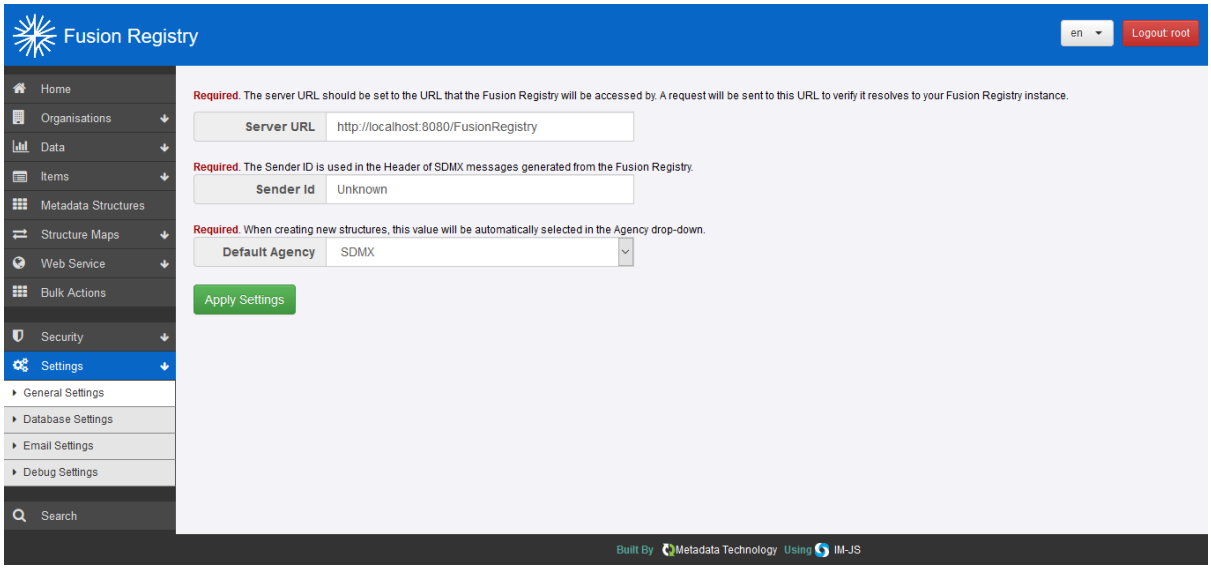

Figure 10 showing the settings tab of Fusion Registry and 'General Settings' selected

The links in the menu allow access to settings for specific sections. Some of these sections have been discussed previously in this document, in the install section:

- General – see section 3.3
- Database – see section 3.2
- Security – see section 3.4

The remaining settings pages are discussed in the following sections.

10.2 Email Settings

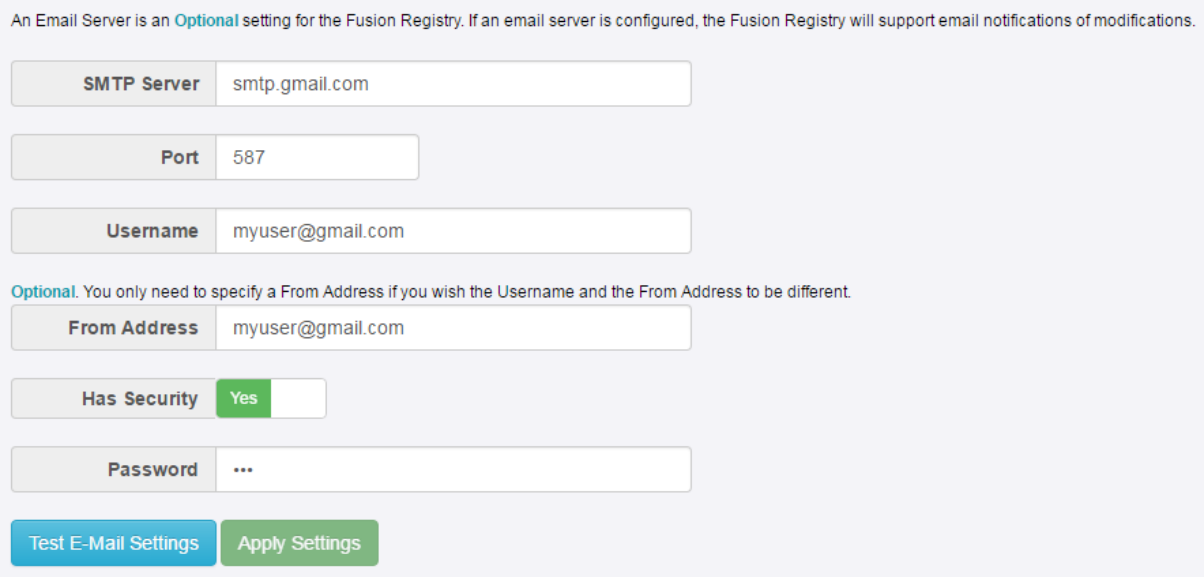

Figure 11 showing the Email Server settings

The Fusion Registry does not require an email server. Iif one is configured then the Fusion Registry will use it for the following:

1. To email reset password details on request.
2. To support users subscribing to changes in the Fusion Registry.

Before the email settings can be applied, you must click the `Test Email Settings` button and send a test email. This is to ensure that the Registry is able to determine if it can communicate with the email server with the specified credentials.

10.3 Debug Settings

This page allows an administrator to specify the debug levels on the Server side of the Registry. Unless you have good reason to modify this setting, it is recommended to set the Server Side Debug set to “Low” or “Warn”.

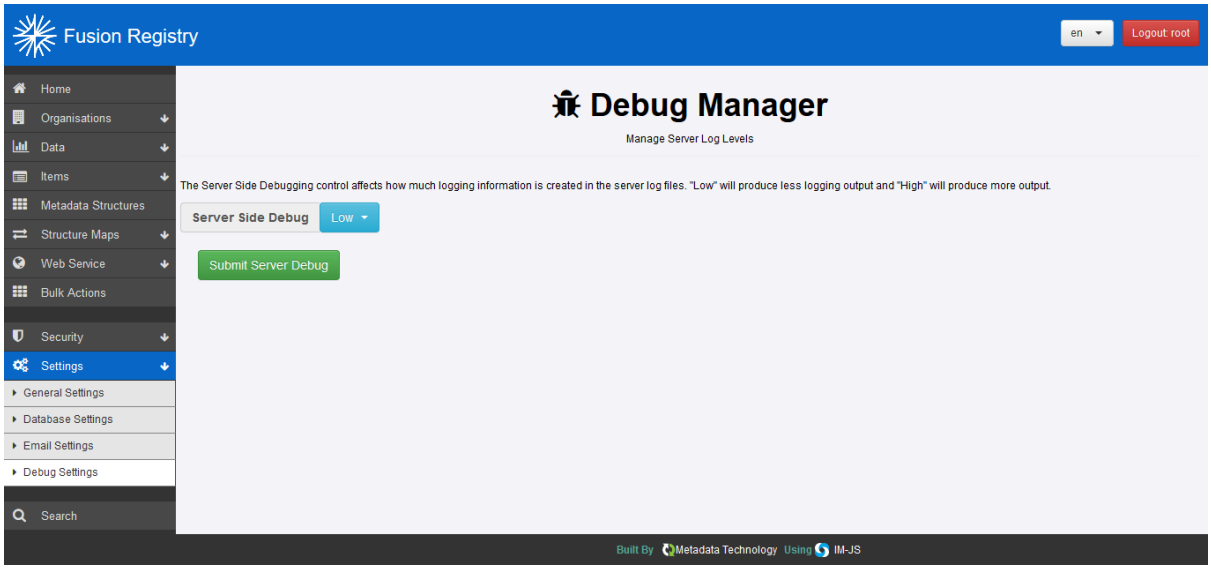

Figure 12 showing the Debug Manager settings page

The Server Side Debugging control affects how much information is stored in the Registry logs on the server side. This pertains to how the Java code is logged by the logging managers. The four values are:

1. Warn – Only log levels of WARN or ERROR are recorded in the logs.
2. Low – the default setting. Only log levels of INFO or above (WARN, ERROR, etc.) are recorded in the logs.
3. Medium – All libraries pertaining to Metadata Technology will be logged at DEBUG level. All other libraries will be at INFO level.
4. High – Everything is logged at DEBUG level, including all third-party libraries.

Please note that logging can introduce a performance impact. This is because logging information is persisted to files and to the database and so almost every interaction with the Registry could be affected.

Modifying this setting is only recommended if there is good reason to do so.

11 Running Apache Tomcat as a Service

11.1 Tomcat as a Service

On Windows environments you may wish to run Apache Tomcat as a service. If you wish to do this there are three important issues to be aware of:

- By default the amount of memory that is allocated to Tomcat as a service will almost certainly not be enough to run Fusion Registry with anything beyond a trivial amount of structures and data. It is recommended that if you wish to run Tomcat as a service, then you increase the "Maximum Memory Pool" value (see section 11.3).
- It may be desirable to explicitly state the location of the Fusion Registry properties file.
- Configuration of Apache Tomcat is not performed by modifications to the setenv.bat file and must be performed by configuring the service.

To configure Tomcat as a service, start the Tomcat Configuration process (on Windows 10, type "Configure Tomcat" from the Windows Start menu).

11.2 Properties file Location

Services on Windows provide the ability to be run as a specified user. This can be changed (see image below) but by default the "Local System account" will be used.

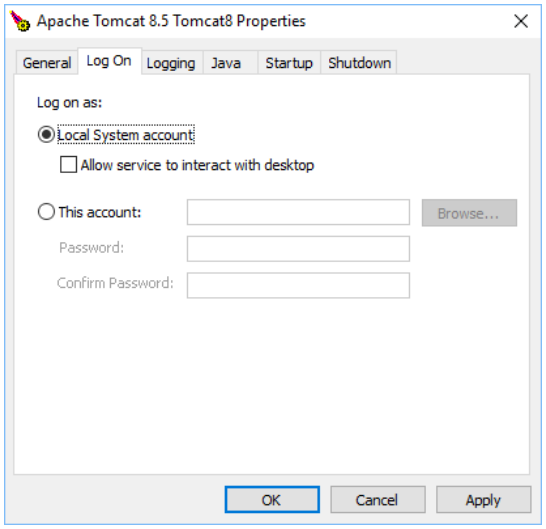

Figure 13 showing the Apache Tomcat Manager "Log On" settings page

If using the "Local System account" then the home directory is a location in the Windows folder itself. This actual value will vary on different version of Windows, but on Windows 10, it may be:

```
C:\Windows\System32\config\systemprofile\FusionRegistry
```

Important note: This location is not accessible to any users except those without Administration access.

This location will be used to store and retrieve Fusion Registry properties. It is probably not desirable to use this location. Modifying this can be achieved in one of two ways:

- Changing the "Log On" user for the Tomcat Service to be another user on the system. This will have the effect of using that user's home directory for the properties file. E.g. if the user

specified is “user1” then the Fusion Registry properties file will probably be located in c:\users\user1\FusionRegistry.

- Providing the argument to specify the location of where the properties file will be read from and written to. This is explained further in section 5.2 and section 11.3 of this document.

11.3 Configuring Apache Tomcat

The file “setenv.bat” cannot be used to configure Apache Tomcat when run as a service. Instead the Apache Configuration Tool has a tab labelled “Java”. From this tab the appropriate settings can be applied.

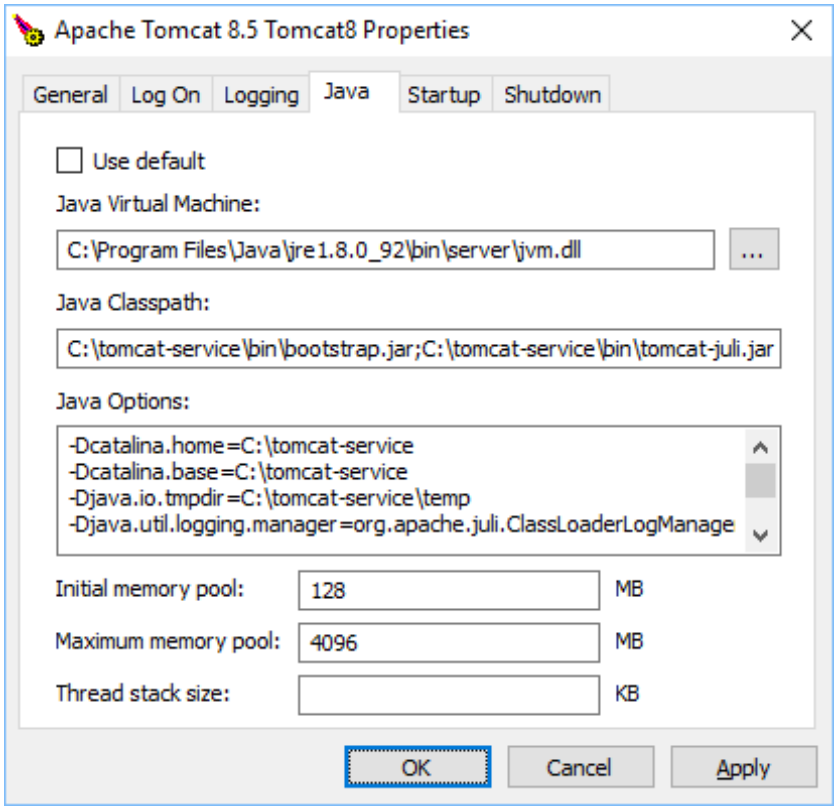

Figure 14 showing the Apache Tomcat Manager Java Properties page

By default, the Initial memory pool Maximum Memory Pool is set to 128Mb and the Maximum memory pool is set to 256Mb. This Maximum Memory Pool will almost certainly not be sufficient to run Fusion Registry so it is recommended to increase this value to at least 2048 Mb. These two settings are the equivalent of settings –Xms and –Xmx in the setenv.bat file.

Other settings, such as “java.library.path” (for SQL Server SSO) must be entered in the input area labelled “Java Options”. Ensure each setting is on its own line. The example below shows typical Apache settings along with the Fusion Registry properties location and the SSO DLL location being set:

```
-Dcatalina.home=C:\tomcat-service
-Dcatalina.base=C:\tomcat-service
-Djava.io.tmpdir=C:\tomcat-service\temp
-Djava.util.logging.manager=org.apache.juli.ClassLoaderLogManager
-Djava.util.logging.config.file=C:\tomcat-service\conf\logging.properties
-DRegistryProperties=file:///c:/dir/AFile.txt
-Djava.library.path=C:\temp\SSO_DLL
```

12 Configuring User Agent

When Fusion Registry performs HTTP communication to other servers (e.g. when performing a registration) the HTTP request header will specify a value for "User-Agent". By default this value will be a string of the form:

FusionRegistry/<version number of the Registry>

For example:

FusionRegistry/9.4.0

If you wish to modify the User Agent that Fusion Registry supplies, you will need to edit a properties file within the Registry. The properties file is called:

metadata.properties

And is located in the directory:

<Tomcat HOME>\webapps\<Web AppName>\WEB-INF\classes

There will be an entry in this file which states the value for user.agent in the following manner:

```
user.agent=FusionRegistry/9.4.0
```

This value can be set to whatever value you wish the User Agent to be. It is permissible to leave this value blank if you require the Registry to not specify a User Agent in HTTP communication. Please note that since this file is within Fusion Registry itself and not in the Settings file (see section 5) if you upgrade Fusion Registry in the future, you will need to make the change again.

It is recommended that you do not modify any of the other values in this file.

13 Recovering a Locked Local Root Account

13.1 Unlocking

If the root account becomes locked, to unlock it you will need access to the underlying database that stores all of the Registry information. In the database there will be a table with the name '*registry_root_security*'. This table contains the following information about the root account: the username; the password (encrypted); the number of times a wrong password can be entered before the account is locked; whether or not the root account is locked.

To unlock a locked root user set the value of the column '*is_locked*' to 0. There is no need to restart the Registry after this change, the root user is now unlocked.

14 Reverting to the Install Wizard

If, for whatever reason, you need to return the Registry to the state of displaying the Install Wizard, then you will need access to the underlying database that stores all of the Registry information. In the database there will be a table with the name '*registry_settings*'. This table contains information about the Registry in a table with column names of 'name' and 'value'. One of the rows in this table will have the name of 'installed.version' and the value will be the version of the Registry.

Deleting this row in the table, committing this change to the database and then restarting the Registry will ensure that on restart that the Registry is displaying the install wizard.