

# FUSION METADATA REGISTRY SETUP GUIDE

FUSION METADATA REGISTRY  
VERSION 10.5.#

## Setup Guide

This guide explains how to install and configure the Registry.

# Contents

---

1	Introduction .....	3
1.1	Scope of this document.....	3
1.2	The Fusion Metadata Registry Software Stack .....	3
1.3	Step 1 - set up the computing environment.....	3
1.4	Step 2 – configure the Java web application environment .....	4
1.5	Step 3 – Extract the war file form the software distribution file .....	4
2	Installation using Tomcat .....	6
2.1	Open Fusion Metadata Registry in a browser .....	6
3	Configure Fusion Metadata Registry .....	7
3.1	Other Configuration Options .....	9
4	Fusion Registry Properties File .....	10
4.1	Introduction.....	10
4.2	Changing the location of the properties file.....	11
4.3	Overriding Server URL Property.....	11
5	Security Roles and Registry Access.....	12
5.1	Overview.....	12
5.2	Registry Security .....	12
5.3	Fusion Security.....	13
5.4	Active Directory .....	14
5.5	Overview.....	14
5.6	Connection Settings .....	14
5.7	Role mapping.....	15
5.8	Role Template.....	16
6	Running Apache Tomcat as a Service .....	17
6.1	Tomcat as a Service.....	17
6.2	Properties file Location .....	17
6.3	Configuring Apache Tomcat .....	18
7	Configuring User Agent.....	19

# Version History

---

Version #	Implemented By	Revision Date	Reason
20191022	Phil Lazarou	22 <sup>nd</sup> Oct 2019	Updated Install Section and Security Section
20200914	V Murrell	14 <sup>th</sup> Sep 2020	Product name change. Removal of unnecessary sections.
20201104	V Murrell	4 <sup>th</sup> Nov 2020	Final review

# 1 Introduction

## 1.1 Scope of this document

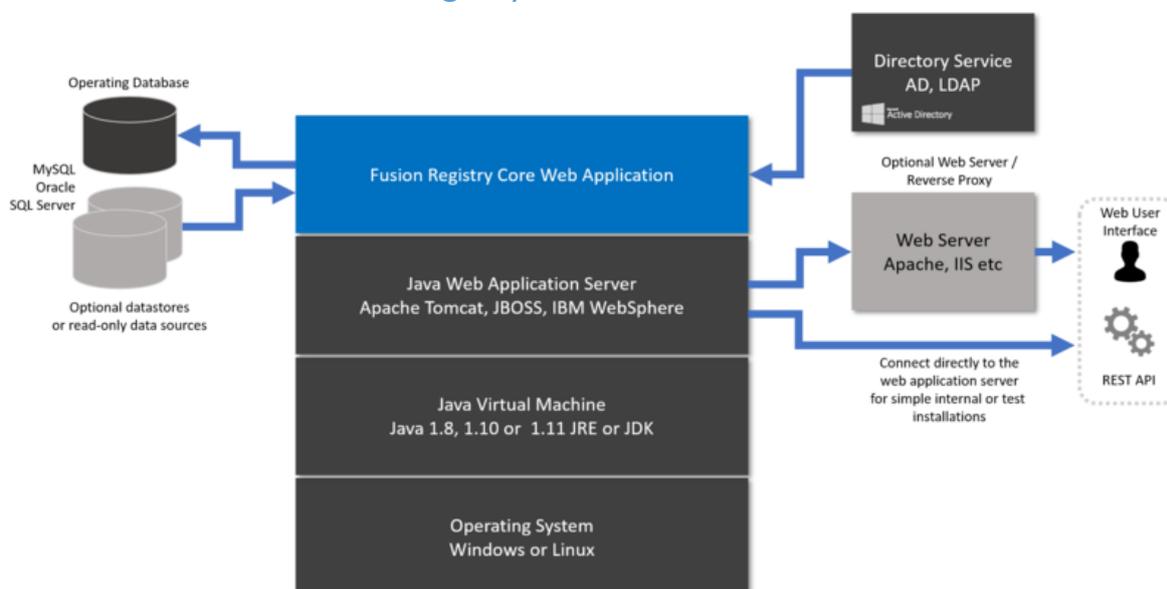
The purpose of this document is to provide an overview of how to install Fusion Metadata Registry in a Java Servlet Container. This document explains the Registry install process, how to configure the Registry and specific advice on configuring aspects of the Java Servlet Container, Apache Tomcat.

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

## 1.2 The Fusion Metadata Registry Software Stack



## 1.3 Step 1 - set up the computing environment

Note that this example is using Apache Tomcat. If you are using a different web application server, refer to its documentation for guidance.

Configuration for Fusion Registry is principally set by creating a setenv.bat or setenv.sh script in the Tomcat bin directory.

Computing Platform	Windows 10 or Linux.
Computer Memory	Minimum of 4GB of memory for a simple installation, but more memory will be required when handling larger datasets.
Java Runtime Environment	Java 1.8 or 1.10 JRE or JDK. Amazon Corretto 8 is a good free-to-use choice.
Database	Fusion Metadata Registry needs a SQL database service to store its metadata content and configuration. MySQL, SQL Server and Oracle are supported. For a quick-start installation, install a local copy of MySQL, although existing database services can be used if you have them.

Web Application Server	The Fusion Registry software runs under the control of a Web Application Server. Options include Apache Tomcat, Jetty, JBOSS and IBM Websphere. Apache Tomcat 8.5 or 9.0 are recommended.
Authentication Directory Service (optional)	The Registry can operate in single-user mode. But if you need to allow multiple users to login with different privileges then an external directory service is required. Microsoft Active Directory or LDAP (OpenLDAP is a free-to-use LDAP implementation) are recommended. Alternatively, Metadata Technology's Fusion Security provides a simple directory service that can be installed alongside Fusion Metadata Registry.

## 1.4 Step 2 – configure the Java web application environment

It is important to override the default Tomcat server memory settings as the default Tomcat settings will not be adequate to run the Registry. 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 Fusion Metadata Registry distribution file contains a setenv.bat (Windows) and setenv.sh (Unix) with the recommended minimum settings configured. These files can be found in the \bin folder. These files can be copied (and optionally modified) to the Tomcat/bin folder before starting the Tomcat instance.

These files set the maximum heap size to 4GB.

Other Parameters can also be included in the setenv batch file as follows:

Parameter	Purpose	Example
-Xmx	Sets the maximum amount of memory that can be allocated to the Java heap. This should be a minimum of 4GB and ideally about 75% of available memory on the machine. Setting too low may result in Java 'out of memory' errors.	-Xmx4GB  sets the maximum heap size to 4GB
-Xms	Sets the initial amount of memory to be allocated to the Java heap. When running in production, there may be a performance benefit to setting this to the same value as -Xmx which avoids the need to grow the heap as memory demand grows.	-Xms128m  sets the initial heap to 128MB
-server	Improves performance by optimising the way the Java Virtual Machine works.	
-DRegistryProperties	System Property: Sets a specific path for the Fusion Registry properties file. If not set, the default location is used	-DRegistryProperties=file:\\opt\tomcats\environments\fusion_registry.properties

## 1.5 Step 3 – Extract the war file form the software distribution file

Fusion Metadata Registry is a Java web application, so deployment and upgrade of the software is different to Linux packages or native Windows applications that come with installation wizards.

The Registry Core server is distributed as a single ZIP archive that contains several files and sub-directories. For instance:

Name	Type	Compressed size	Password ...	Size	Ratio	Date modified
bin	File folder					19/12/2019 10:27
DataSamples	File folder					19/12/2019 10:27
Documentation	File folder					19/12/2019 10:27
password reset	File folder					19/12/2019 10:27
10.0-Changelog.txt	Text Document	37 KB	No	130 KB	72%	19/12/2019 10:40
FusionRegistry.war	WAR File	143,032 KB	No	143,238 KB	1%	19/12/2019 10:27
README_Important.bt	Text Document	1 KB	No	1 KB	29%	04/06/2019 17:03

Save the contents of Zip file as you may need them later.

## 2 Installation using Tomcat

Fusion Metadata Registry consists of a single .war file called FusionRegistry.war.

Web application server products have different procedures for deploying applications. The procedure here applies to Apache Tomcat. If using a different web application server, you will need to refer to the appropriate product guide.

1. If the Tomcat is already running, stop it.
2. Copy the war into the Tomcat's **webapps** folder for example:  
  
C:\tomcat\FMR\Registry-8080\webapps
3. Start the Tomcat - in the **bin** folder you will find a file 'startup.bat' which should be double clicked.

When the Tomcat is started, the war file will be unpacked, and a properties file will be created and, by default, will be placed into the 'home' directory. A typical location is shown in the example below.

C:\Users\Val\MetadataTechnology\FusionRegistry

When the Tomcat starts, it will expect to find the properties file in this location. If this location is not suitable, it is possible to move the properties file as discussed in the section '**Step 2 – configure the Java web application environment**' above.

Please check the Tomcat log files to ensure that Fusion Metadata Registry has deployed correctly.

### 2.1 Open Fusion Metadata Registry in a browser

Once the Tomcat has successfully started, open a Web Browser (excluding Internet Explorer which is no longer supported) and enter the URL, for example:

- <http://localhost:8080/FusionRegistry>
- <http://myserver.mydomain.local:8080/FusionRegistry>
- <http://192.168.1.27:8080/FusionRegistry>

The hostname is dependent on the configuration of the machine on which Fusion Metadata Registry has been installed. The port 8080 is the Tomcat default and may be different if you have altered the Tomcat settings.

The first of the two configuration pages will open as detailed below.

### 3 Configure Fusion Metadata Registry

**Tab 1 - Database**

The screenshot shows the 'Fusion Registry 10 - Install' configuration interface. At the top, there is a blue header with the Fusion Registry logo and the text 'Fusion Registry 10 - Install'. Below the header, there are two tabs: '1. Database Connection' (active) and '2. Server Settings'. The main content area contains a form for configuring the database connection. A note states: '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.' The form includes the following fields: 'Database Type' (a dropdown menu set to 'MySQL'), 'Server' (text input 'localhost'), 'Port' (text input '3306'), 'Schema' (text input 'de-structures-06-08'), 'Userid' (text input 'root'), and 'Password' (password input field with masked characters). A green 'Apply Settings' button is located at the bottom left of the form.

This tab allows you to provide the connection details to your chosen database service which will be used as the main operating database.

Options available are:

- MySQL
- Oracle
- SQL Server
- Custom

The schema must already exist but can be empty in which case the Registry will create all the necessary tables. If you specify an existing Registry schema here, the service will attempt to load the content and settings within.

If your database service requires special parameters such as authentication details, use the Custom Database Type option which allows the JDBC connection string to be set.

To move to the next tab, click Apply Settings. Note that if there are errors found, the system will identify issues and you must correct before you will be able to continue the configuration process.

## Tab 2 – Server Settings

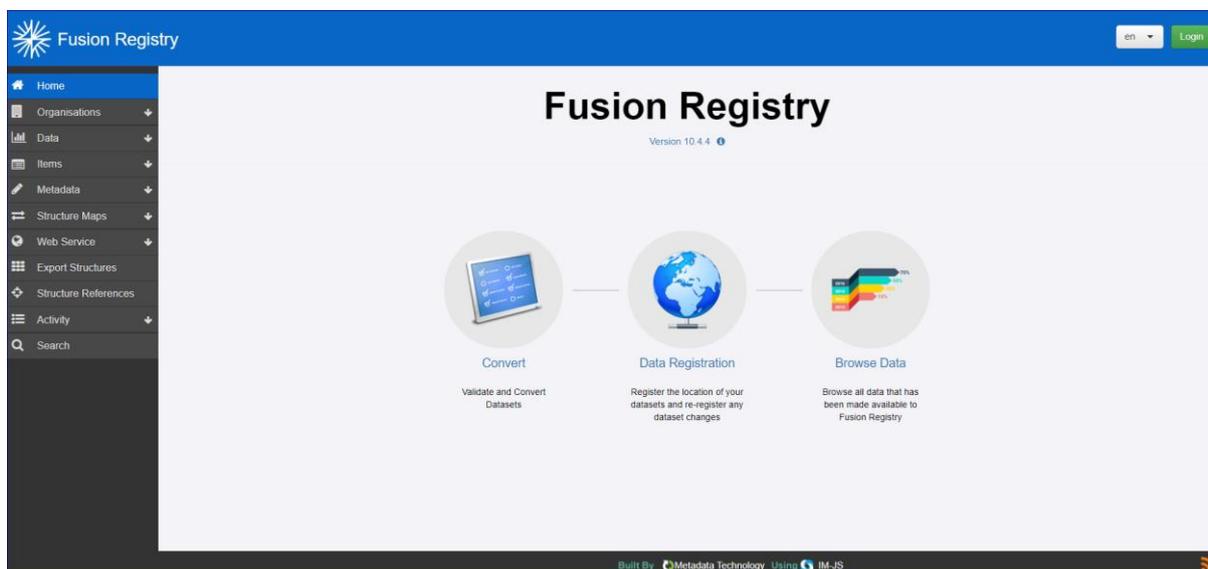
The screenshot shows the 'Server Settings' tab in the Fusion Registry 10 installation wizard. It features a 'Required' section with a note about the server URL. Below this are input fields for 'Server URL' (http://localhost:8080/FusionRegistry), 'Username' (Amanda56), 'Password', 'Repeat Password', and 'Max Login Attempts' (-1). Navigation buttons for 'Previous' and 'Finish' are at the bottom right.

The server URL is the full path, for example <http://localhost:8080/FusionRegistry/>.

The Username and Password is for top level access to the Registry known as the 'root user' account.

You can also set the maximum number of login attempts here, -1 is unlimited.

Click Finish, the Registry will open.



### 3.1 Other Configuration Options

You can further configure your Fusion Metadata Registry as follows:

- Add your own logo, favicon and change the colour
- Configure Audit settings
- Configure Debug settings
- Configure an Email Server
- Include a Support email address
- Include a Support URL
- Change the Default Agency

These options are discussed in the Fusion Metadata Registry's wiki [which you can find here](#).

## 4 Fusion Registry Properties File

### 4.1 Introduction

Whilst configuration of Fusion Metadata Registry is performed via the Web User Interface, database connection details are stored in a local properties file.

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

The properties file is called:

*fusion\_registry.properties*

By default, the 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, 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 Registry start-up, the properties file will be loaded 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]
```

## 4.2 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 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 the Registry will NOT start if this value is incorrect or if this file cannot be created.

## 4.3 Overriding Server URL Property

When deploying multiple load-balanced Fusion Metadata 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.

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.

## 5 Security Roles and Registry Access

### 5.1 Overview

The Fusion Metadata Registry has 3 distinct user roles, these are:

1. Root
2. Admin
3. Agency

Users with Root and Admin roles can access all parts of the Registry application. Only the Root user can change the Root user credentials.

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

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

### 5.2 Registry Security

The Registry provides the following security mechanisms for authentication:

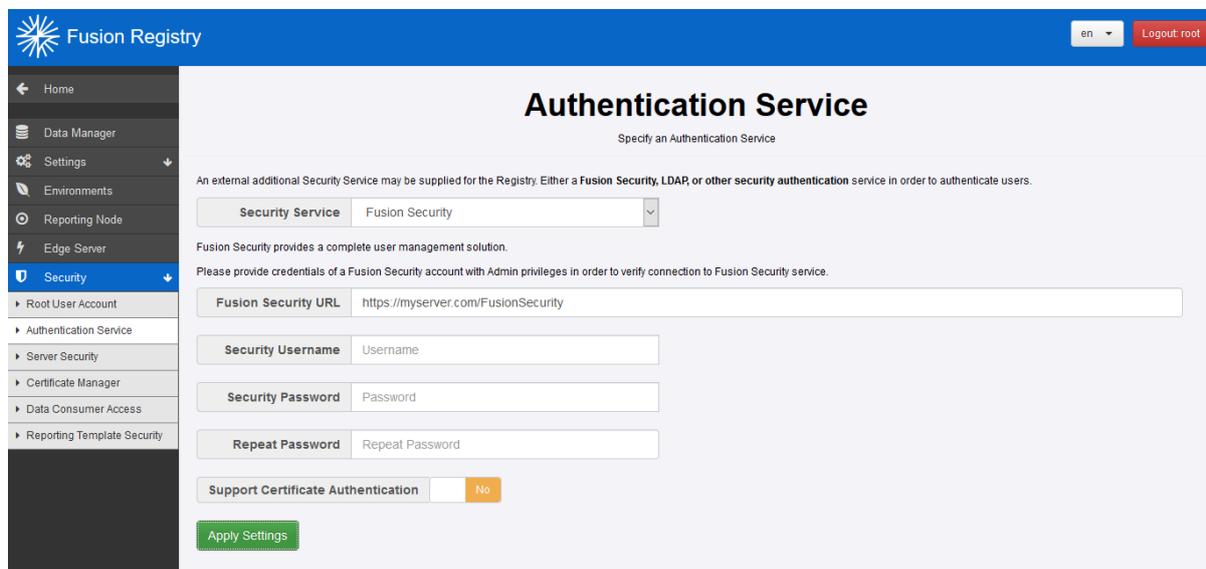
1. **Local Security** – Local security authenticates the user within the Fusion Metadata Registry application. Only one root user account exists, and this user has unrestricted access to the whole application. The local root credentials are encrypted and stored in the Registry database. Local Security 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 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 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.

## 5.3 Fusion Security

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



The screenshot shows the Fusion Registry Administration interface. The top navigation bar includes the Fusion Registry logo, a language dropdown set to 'en', and a 'Logout: root' button. A left sidebar contains navigation options: Home, Data Manager, Settings, Environments, Reporting Node, Edge Server, Security (highlighted), Root User Account, Authentication Service, Server Security, Certificate Manager, Data Consumer Access, and Reporting Template Security. The main content area is titled 'Authentication Service' and includes the subtitle 'Specify an Authentication Service'. It contains a dropdown menu for 'Security Service' set to 'Fusion Security', a text input for 'Fusion Security URL' with the value 'https://myserver.com/FusionSecurity', and three password input fields for 'Security Username', 'Security Password', and 'Repeat Password'. A checkbox for 'Support Certificate Authentication' is present and unchecked. An 'Apply Settings' button is located at the bottom of the form.

Figure 1 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 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 Metadata Registry.

To learn more about Fusion Security please contact us by emailing [info@metadatatechnology.com](mailto:info@metadatatechnology.com).

## 5.4 Active Directory

### 5.5 Overview

Active Directory may be used as the Authentication and Authorisation provider for Fusion Metadata Registry. This requires an Active Directory server running on a network that the Registry can communicate to. Fusion Metadata 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.

### 5.6 Connection Settings

To establish the connection between Fusion Metadata 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.

**Authentication Service**  
Specify an Authentication Service

An external additional Security Service may be supplied for the Registry. Either a Fusion Security, Active Directory or LDAP authentication service in order to authenticate users.

Security Service	Active Directory
Protocol	ldap://
URL	77.68.55.999
Active Directory Domain	example.domain
Base Group DN (leave blank for root)	OU=QA,OU=MT,DC=example,DC=domain
User Search Filter	(&(objectClass=user)(userPrincipalName={}))

Apply Settings

Figure 2 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.

## 5.7 Role mapping

To setup a mapping, select the Role Mappings option from the Security menu to open the Role Mapping page.

Role Mappings		
Map Active Directory/LDAP Roles to SDMX Organisations		
AD / LDAP Role	Mapped Organisation	Agency
YADMIN	Administrator	N/A
ACY_ECB	Agency	ECB
ACY_WB	Agency	WB
ACY_ECB_SUB	Agency	ECB.SUB
ACY_ESTAT	Agency	ESTAT

The example above shows that an Administration role has been defined together with several Agency roles.

The AD/LDAP role is the Group defined in AD as shown below for the Administrator role.

Name	Type	Description
WB_DC	Security Group...	Not Template Mapping
WB_DP	Security Group...	Not template Mapping
ACY_ECB_SUB	Security Group...	Not template mapping
YADMIN	Security Group...	Not template mapping
METATECH_DP	Security Group...	Not using Template Mapping
ESTAT_DP	Security Group...	Not using template mapping
PRL_FR_GOU_FFT	Security Group...	PHIL's
ACY_GCCSTAT	Security Group...	Template Mapping
ACY_GCCSTAT_SUB1	Security Group...	Template Mapping
ACY_OECD	Security Group...	Template Mapping

The User is defined in AD and placed in the relevant AD Group.

To add a Mapping, select the Add Mapping button. This will open a list of Organisations as defined in the Registry. The owning Agency appears – example [ECB:SUB] followed by the Agency Name – example “Elaine S ECB:Member of ACY-ECB\_SUB”.

**Role Mapping**

AD/LDAP ROLE:  AD ROLE:

Administrator

**Agencies**

[SDMX] SDMX

[ESTAT] ESTAT : Member of ACY\_ESAT (Template Mapping)

[OECD] Organisation for Economic Co-operation and Development (OECD)

[WB] World A Bank : Member of ACY\_WB (template Mapping)

[IAEG] Inter Agency Group

[A9\_7\_0] Version 9.7 New Agency

[ECB] Eric ECB: Member of ACY\_ECB

[A10\_3\_1] FR-4483

[ECB.SUB] Elaine S ECB : Member of ACY\_ECB\_SUB

**Data Providers**

[ECB.RAYMOND] Ray DP:ECB : Member of ECB\_DP

[ESTAT JAMES] James Provider : Member of ESTAT\_DP

To map, click to select and then click the Assign button.

## 5.8 Role Template

If no Role Mappings are defined, the Registry will communicate to Active Directory using a 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 create a group that permits Administrator access to the Registry, the group needs to be named "Administrators" (this is not case sensitive).

## 6 Running Apache Tomcat as a Service

### 6.1 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 6.3).
- 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).

### 6.2 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.

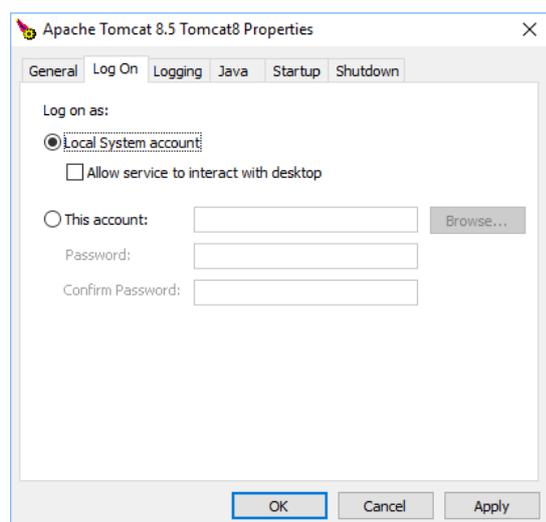


Figure 3 showing the Apache Tomcat Manager “Log On” settings page

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 0 and section 6.3 of this document.

### 6.3 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.

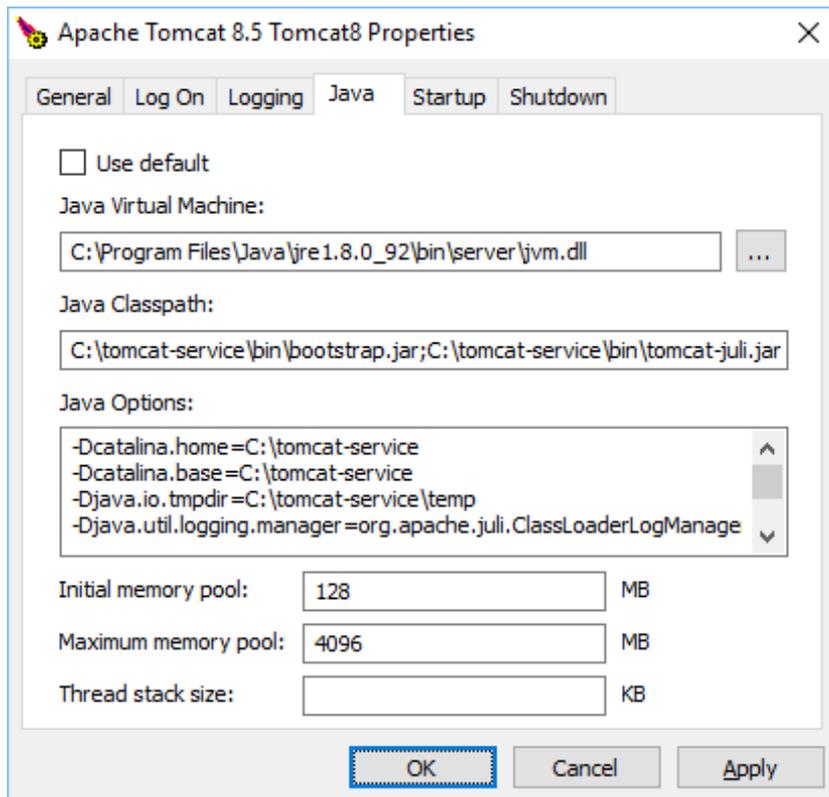


Figure 4 showing the Apache Tomcat Manager Java Properties page

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 the 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 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
```

## 7 Configuring User Agent

When Fusion Meta Data 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 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 the Registry itself and not in the Settings file, if you upgrade Fusion Metadata 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.**

---