



## Fuseki Server Installation

Related task of the project (Task # and full name):	Task 4.3 Ontology standard and Metadata
Author:	Sachin Deshmukh
Prepared by:	Sachin Deshmukh
Approved by:	Richard Kaye

## Table of contents

1	Definitions .....	3
1.1	Acronyms and Abbreviations .....	3
2	Introduction.....	4
2.1	Goals .....	4
3	Installation Requirements .....	4
3.1	System Requirement.....	4
3.2	Prerequisites .....	4
4	Fuseki installation and Configurations .....	4
4.1	Fuseki Server Setup:.....	4
4.2	Fuseki server system layout:.....	5
4.3	Fuseki configuration and service assembler: .....	5
4.4	Fuseki Server configuration as a Service:.....	6
4.5	Fuseki Server configuration to run as standalone Server: .....	7
5	Conclusion: .....	8
6	Appendix.....	9
6.1	FusekiConfig.ttl .....	9
6.2	Fuseki file .....	10
7	Document Revision History .....	11

# 1 Definitions

## 1.1 Acronyms and Abbreviations

Acronym	Definition
RDF	Resource Description Framework
SPARQL	RDF query language
JENA	A free and open source Java framework (maintained by Apache) for building Semantic Web and Linked Data applications
Fuseki	A component within the JENA framework, comprising a SPARQL server providing support for REST-style queries over HTTP
<SERVER_NAME>	The server where Fuseki is installed
<FUSEKI_HOME>	Fuseki server installation location
<FUSEKI_BASE>	Fuseki root of runtime area
<DATASET_SERVICE>	Dataset service name (associated with RDF)
<FUSEKI_URL>	Fuseki server URL e.g. http://<SERVER_NAME>:3030

## 2 Introduction

In order to provide annotation support in the Model Repository, there needs to be an RDF store which is pre-populated with metadata information. Access to this RDF store is managed using a Fuseki server accessible via the SparQL protocol over HTTP. This document provides instructions for setting up such a Fuseki server.

A Fuseki instance has been set up and configured on the DDMoRe Uppsala server and the same toolset and instructions are considered as guidelines for this document.

### 2.1 Goals

The aim of this document is to enable a user to install a new Fuseki server and configure it to access a local RDF store using the SparQL HTTP protocol.

## 3 Installation Requirements

### 3.1 System Requirement

Tool	Version Used
JDK	1.7
Apache-JENA-Fuseki	2.0.0
Apache-JENA	2.13.0

As JDK 1.7 is the standard java used across DDMoRe, we are using compatible JENA and Fuseki versions. Note that the latest JENA and Fuseki versions are only supported under Java8.

### 3.2 Prerequisites

- Authorisation details to access the <SERVER\_NAME> environment and the ability to add/update configurations and execute services.
- RDF store with Dataset service information to configure in Fuseki server. This can be performed later but then the server will need to be restarted.

## 4 Fuseki installation and Configurations

In this document a Fuseki installation on a Linux server host will be described. If the server host is Windows-based then some of the changes are not required and will be mentioned in the notes.

### 4.1 Fuseki Server Setup:

Apache JENA framework related information and tools are available at [this](http://jena.apache.org) (jena.apache.org) website.

To setup Fuseki-2, its archive needs to be downloaded from [this](#) location and unzipped at the destination location.

The Fuseki server can be installed as a ‘standalone’ server or an ‘always on’ service. Please see [this](#) page on the Fuseki documentation website for details of the configuration. This documentation page covers installing Fuseki as a standalone server as well as a service.

Please note that the Uppsala server has been configured ‘as a service’ but local setups are standalone server setups. This document describes both of these setups but the ‘as a service’ approach is described in detail.

## 4.2 Fuseki server system layout:

This section describes how the server file system is organised and what the options are to configure the server. The general details are specified at [this](#) page. The default file system settings for Linux host are as below.

Mode	Environment Variable	Default Setting
Service	<FUSEKI_HOME>	/usr/share/fuseki
	<FUSEKI_BASE>	/etc/fuseki
Webapp	<FUSEKI_HOME>	N/A (Files in the Fuseki .war file)
	<FUSEKI_BASE>	/etc/fuseki
Standalone	<FUSEKI_HOME>	Current directory
	<FUSEKI_BASE>	\${FUSEKI_HOME}/run/

Depending upon server mode, the values for FUSEKI\_HOME and FUSEKI\_BASE should be specified in the relevant server configuration file. Default values are loaded from the file /etc/default/fuseki, if it exists.

Note: If <FUSEKI\_BASE> is same as <FUSEKI\_HOME> then care needs to be taken while upgrading Fuseki so as not to delete server deployment/configuration files and directories.

## 4.3 Fuseki configuration and service assembler:

The Fuseki server can be configured using a configuration file. This configuration file is an RDF graph with one server description and a number of data services. Each data service offers a number of endpoints over the dataset.

To illustrate with an example, the configuration file is added to appendix of this document as [FusekiConfig.ttl](#) (Please follow the link to see the complete file).

The detailed explanation of the sections of this configuration file can be found at [this](#) JENA-Fuseki documentation page.

Following is the example of the DDMoRe metadata dataset from the configuration file.

```
## ----- ddmore Metadata dataset -----
<#service2> rdf:type fuseki:Service ;
fuseki:name "ddmore-metadata" ;
fuseki:serviceQuery "query" ;
fuseki:serviceReadGraphStore "get" ;
fuseki:dataset <#ddmore-md-dataset1> ;
.

<#ddmore-md-dataset1> rdf:type ja:RDFDataset ;
ja:defaultGraph <#ddmore-md-graph1> ;
.

<#ddmore-md-graph1> a ja:InfModel;
rdfs:label "DDMoRe-MetaData-1" ;
ja:baseModel
[ a ja:MemoryModel ;
# ddmore SCHEMA
ja:content [ja:externalContent <file:data/ddmore-store-<version>.ttl]& ] ;
] ;
.
```

The service name configured is “ddmore-metadata” and the dataset is loaded from the file system. The dataset file in this example is ddmore-store-<version>.ttl, where <version> is the appropriate dataset version. Note that at the time of writing, the dataset <version> is ‘v1.2.3-prod’

These datasets are stored in the GitHub project at [this](#) location and can be downloaded as required.

#### 4.4 Fuseki Server configuration as a Service:

1. In order to configure Fuseki as a service, the appropriate file system layout as described in section 4.2 is required.

The example ‘fuseki’ file is added to Appendix 5.1 [Fuseki file](#).

The important updates to the file are for the following properties,

- (a) JAVA – Java installation location
- (b) FUSEKI\_HOME – Fuseki installation location
- (c) FUSEKI\_BASE - The runtime area. Location for log files, system files, local config
- (d) FUSEKI\_CONF – Fuseki configuration file location

The file pointed to by FUSEKI\_CONF is the configuration file which has the dataset services configured. It needs to exist and have the appropriate permissions.

2. The config file at ‘/etc/init.d/fuseki’ needs to be amended. Please note that this config file imports configurations specified at ‘/etc/default/fuseki’ and then extends them further.

The following lines from the configuration should be commented out as below:

```
#if [ ! -z "$FUSEKI_HOME" ]
#then
# FUSEKI_ADDITIONAL_ARGS=( --home "$FUSEKI_HOME" )
#fi
```

This is because the '--home' option is not valid as described in the Fuseki help documentation.

3. '\$FUSEKI\_HOME/run/shiro.ini' updates

As part of the initial server execution, the setup has created the directories and files required. The 'run' directory and its files are created and referred to when the server is running.

When server is started from outside using a URL (eg http://<SERVER\_NAME>:3030), the server manager is not able to access the datasets and related management tools.

The following entry in "shiro.ini" needs to be changed in order to enable access to the dataset from outside. This is required in order to execute Sparql queries over HTTP.

```

...
[urls]
## Control functions open to anyone
    /$/status = anon
    /$/ping    = anon
## and the rest are restricted to localhost.
    /$/** = localhost
...
Here we need to remove this restriction to allow 'only localhost' to allow 'everyone' as below,

    /$/** = anon
  
```

4. Starting the service:

- It should automatically (re-)start following a machine reboot.
- From the command line:  
*>sudo service fuseki start*

5. Monitoring the service:

- From the command line:  
*>sudo service fuseki status*

6. Stopping the service:

- From the command line:  
*>sudo service fuseki stop*

#### 4.5 Fuseki Server configuration to run as standalone Server:

- Please ensure that configuration file (e.g. FusekiConfig.ttl) is in an accessible directory.
- Please confirm that the data file specified in the configuration file does exist and includes the correct data.
- Start the server using following command:

```
>fuseki-server --config=<relativePath>/fusekiConfig.ttl

[] Server      INFO  Fuseki 2.0.0 [date:time]
[] Config      INFO  FUSEKI_HOME=<path>/apache-jena-fuseki-2.0.0/.
[] Config      INFO  FUSEKI_BASE=<path>/apache-jena-fuseki-2.0.0/run
[] Servlet     INFO  Initializing Shiro environment
[] Config      INFO  Shiro file: file://<path>/apache-jena-fuseki-2.0.0/run/shiro.ini
[] Config      INFO  Configuration file: fusekiConfig.ttl
[] Builder     INFO  Service: :service1
[] Builder     INFO  Service: :service2
[] Config      INFO  Register: /ds
[] Config      INFO  Register: /ddmore-metadata
[] Server      INFO  Started [date:time] BST on port 3030
```

- Once server has started successfully, the Fuseki server manager and other services can be accessed at default port 3030. If this port is already in use, the port can be configured in the Fuseki server to another suitable port.

The Fuseki server manager can be accessed at [http://<SERVER\\_NAME>:3030](http://<SERVER_NAME>:3030)

- The dataset services available for service configuration are accessible as below,

The screenshot shows the Apache Jena Fuseki web interface. At the top, there are navigation links for 'dataset', 'manage datasets', and 'help'. Below this, a dropdown menu shows the selected dataset as '/inf1'. There are buttons for 'query', 'upload files', 'edit', and 'info'. The main content area is titled 'Available services' and lists the following endpoints:

- Graph Store Protocol (Read): <http://localhost:3030/inf1/get>
- HTTP Quads: <http://localhost:3030/inf1/>
- SPARQL Query: <http://localhost:3030/inf1/query>

Below the services, there is a 'Statistics' section with a table:

Name	Overall	Overall good	Overall bad	Graph Store Protocol (Read)	SPARQL Query
/inf1	2	2	0	0	1 (0 bad)

At the bottom, there is a 'Dataset size' section with a note: 'Note this may be slow and impose a significant load on large datasets: [count triples in all graphs](#)'. Below that is an 'Ongoing operations' section with a note: 'TBD. Will list any long-lasting operations that are ongoing or recently completed, e.g. backups.'

## 5 Conclusion:

A Fuseki server has been successfully configured as a Service on the Uppsala server (located at <http://repository.ddmore.eu:3030/index.html>) and also as a local standalone server created for development using these configurations.



## 6 Appendix

### 6.1 FusekiConfig.ttl

```

### Fuseki Configuration file with metadata dataset configured.

@prefix :      <#> .
@prefix fuseki: <http://jena.apache.org/fuseki#> .
@prefix rdf:   <http://www.w3.org/1999/02/22-rdf-syntax-ns#>.

@prefix rdfs:  <http://www.w3.org/2000/01/rdf-schema#>.
@prefix tdb:  <http://jena.hp1.hp.com/2008/tdb#>.
@prefix ja:   <http://jena.hp1.hp.com/2005/11/Assembler#>.

[] rdf:type fuseki:Server ;

    fuseki:services (
      <#service1>
      <#service2>
    ) .

### -----
### Updatable in-memory dataset.

<#service1> rdf:type fuseki:Service ;
  # URI of the dataset -- http://host:port/ds
  fuseki:name "ds" ;
  fuseki:serviceQuery "sparql" ;
  fuseki:serviceQuery "query" ;
  fuseki:serviceUpdate "update" ;
  fuseki:serviceUpload "upload" ;
  fuseki:serviceReadWriteGraphStore "data" ;
  fuseki:serviceReadGraphStore "get" ;
  fuseki:dataset <#emptyDataset> ;
  .

### In-memory, initially empty.
<#emptyDataset> rdf:type ja:RDFDataset .
### ----- ddmore Metadata dataset -----
<#service2> rdf:type fuseki:Service ;
  fuseki:name "ddmore-metadata" ;
  fuseki:serviceQuery "query" ;
  fuseki:serviceReadGraphStore "get" ;
  fuseki:dataset <#ddmore-md-dataset1> ;
  .

<#ddmore-md-dataset1> rdf:type ja:RDFDataset ;
  ja:defaultGraph <#ddmore-md-graph1> ;
  .

<#ddmore-md-graph1> a ja:InfModel;
  rdfs:label "DDMoRe-MetaData-1" ;
  ja:baseModel
    [ a ja:MemoryModel ;
  # ddmore SCHEMA
    ja:content [ja:externalContent <file:data/ddmore-store-v1.1.1.ttl>] ;
    ] ;
  .

```

## 6.2 Fuseki file

```
# Configuration
# -----
# Default values are loaded from /etc/default/fuseki, if it exists.
#
# JAVA
#   Command to invoke Java. If not set, java (from the PATH) will be used.
JAVA=/usr/bin/java
# JAVA_OPTIONS
#   Extra options to pass to the JVM.
# FUSEKI_HOME
#   Where Fuseki is installed. If not set, the script will try
#   to guess it based on the script invocation path.
FUSEKI_HOME=/opt/apache-jena-fuseki-2.0.0
# FUSEKI_BASE
#   The root of the runtime area - logs files, system files, local
#   configuration.
#   Defaults to /etc/fuseki.
FUSEKI_BASE=${FUSEKI_HOME}
# FUSEKI_RUN
#   Where the fuseki.pid file should be stored. It defaults
#   first available of /var/run, /usr/var/run, and /tmp if not set.
# FUSEKI_PID
#   The FUSEKI PID file, defaults to $FUSEKI_RUN/fuseki.pid
# FUSEKI_ARGS
#   The arguments to pass to the Fuseki server on the command line. Defaults
#   to:
#           # if FUSEKI_CONF is not set
#   --config=$FUSEKI_CONF           # if FUSEKI_CONF is set
# FUSEKI_START
#   Path to the jar file. Defaults to $FUSEKI_HOME/fuseki-server.jar
# FUSEKI_CONF
#   The Fuseki configuration file, usually in RDF Turtle notation.
FUSEKI_CONF=$FUSEKI_HOME/fusekiConfig.ttl
```

## 7 Document Revision History

Version	Author	Date	Description
1.0	Sachin Deshmukh	26-Aug-2016	Initial version