|  |  |  |
| --- | --- | --- |
| Author: | Brian McBride |  |
| Contributors: |  |  |
| Review: |  |  |

Setting Up a CIM Client and Server

Version 0.2 ()  
03 February 2015

# Introduction

This document contains instructions for configuring and setting up a client and server for the Sedgemoor Community Impact project. It is intended to be used by the Sedgemoor IT group for configuring and maintaining these systems.

The server is the machine that holds the data in linked data form and makes it available on the web. The client is the data collection and management machine. It holds the source data as an Excel spreadsheet and runs scripts to translate the spreadsheet data to RDF format for publishing and to upload the RDF data to the server.

The server is expected to be a 64 bit machine running a Windows server operating system. A typical server for a modest sized database will have 8GB of RAM. This should be sufficient for a modest sized database. For small lightly loaded servers, 4GB of RAM may be sufficient. Use of an SSD can make a significant difference to the performance of a server, especially if the data is large for the amount of memory available.

Systems such as described here are often run on Linux rather than windows. The setup described in this document is a modification of a Linux configuration for Windows. This configuration can likely be improved by modifying it to conform to Windows conventions and to use Windows specific replacements for components such as the Apache web server.

The client is expected to be a Windows desktop machine. In principle the same machine could be used as both server and client, provided the server is able to run Excel.

## General considerations

In these instructions, some choices, such as which directory in which to install things, are left to the installer. These directory path names are then referred to by a name in capitals beginning with a ‘$’ character. For example, the variable $FUSEKI\_HOME refers to the directory in which Fuseki is installed.

Some of the configuration files contain references to these directories and must be edited to replace the variable names with the actual path corresponding to that name. The instructions indicate which files need editing.

Some of the configuration files and command line parameters are interpreted by Java programs. File paths are generally better written using ‘/’ characters as a path separator rather than the convention on Windows which is to use a ‘\’. When such paths are used at a command prompt, it may be necessary to enclose the path in ‘”’ characters to prevent the command interpreter misinterpreting them.

When representing path names in configuration files and command line arguments, treat them as being case sensitive.

In at least one place in the configuration, a file path is written as a file: URL. On Windows, file paths that contain drive letters, when expressed as a file: URL should be written as file:///{drive-letter}:/{path-within-drive}.

The variable $SERVER refers to the domain name or IP address of the server.

# Installing the Software

The software is primarily Java based. The software components that need to be installed are:

* Java – a platform independent programming language
* Fuseki – an RDF triple store. Think of this as the system database.
* Tomcat – this is a web application container. It exists to provide an environment for Elda
* Elda – enables the system to respond to data requests from the web. It provides a configurable Web based API to access the data
* Apache – provides a robust secure Web facing interface for Elda and Fuseki. Apache’s job is to limit access from the web to public services, to provide caching and to control the rate at which requests are passed to Elda and Fuseki to prevent overload.

The following discussions describe how to install each of the components in turn and to perform a rudimentary test that the component is working.

A ZIP file containing configuration files, content etc to be installed has been supplied. This will be needed to complete the installation.

## Installing Java

There are a number of implementations of Java available. On Windows, it is recommended to install the implementation provided by Oracle.

There are a number of variants of Java available with different characteristics. Whilst other variants may work, it is recommended to install the Standard Edition JDK. At the time of writing, an installer could be downloaded from <http://www.oracle.com/technetwork/java/javase/downloads/index.html>.

Java version 7 or later is required.

Follow the instructions provided by Oracle for installation.

Verify the installation by entering the command

java –version

should print out the version of java installed.

## Installing Fuseki

Fuseki is the system database. It is implemented in Java. To install it, carry out the following steps:

1. Download a zip of the latest version from <http://jena.apache.org/download/index.cgi>
   1. At the time of writing the latest version was jena-fuseki-1.1.1-distribution.zip
2. Decide where the Fuseki software should go in the file system. This will be referred to as $FUSEKI\_HOME. On Linux this might be /usr/share/fuseki.
3. Unzip the contents of the Fuseki zip file into that directory.
4. Verify that the file $FUSEKI\_HOME\fuseki-server.jar exists. If it is not in the directory $FUSEKI\_HOME then the directory structure is likely not correct.

To verify that Fuseki will start, start it with its built in test database.

1. At a command prompt:
   1. cd $FUSEKI\_HOME
   2. java –jar fuseki-server.jar –config config.ttl
2. use a browser to access <http://$SERVER:3030/>
   1. the Fuseki welcome screen should be displayed providing access to a small test dataset about Harry Potter books
3. control-c at the command prompt to stop the running fuseki process

Decide which user will run the Fuseki database. This can be admin user, but since Fuseki is an application process it may be best run as an ordinary user. This user will be referred to as $FUSEKI\_USER. This user is typically ‘fuseki’.

Next, set up the database and files to be used by the system. Decide in which directories the Fuseki configuration, database and log files will be held. These will be referred to as $FUSEKI\_CONFIG, $FUSEKI\_DATABASES and $FUSEKI\_LOG respectively. The fuseki user will need read access to $FUSEKI\_CONFIG and read/write access to the other directories. Decide also what that actual database will be called. This will be referred to as $DATASET. Do the following:

1. Create the directories $FUSEKI\_CONFIG, $FUSEKI\_DATABASES/$DATASET and $FUSEKI\_LOG
2. Ensure $FUSEKI\_USER read access to $FUSEKI\_CONFIG and read/write access to to the other directories
3. Copy the provided fuseki-config/config.ttl to $FUSEKI\_CONFIG/config.ttl
4. Edit $FUSEKI\_CONFIG/config.ttl to replace $ variables with their actual values
5. Copy the supplied fuseki-config/log4j.properties file $FUSEKI\_CONFIG/log4j.properties
6. Edit $FUSEKI\_CONFIG/log4j.properties to replace $ variables with their actual values

Run fuseki by entering, as $FUSEKI\_USER the following command at a command prompt :

java -Xmx1024M \  
 "-Dlog4j.configuration=file:$FUSEKI\_CONFIG/log4j.properties" \  
 -jar "$FUSEKI\_HOME/fuseki-server.jar" \  
 --config "$FUSEKI\_CONFIG/config.ttl"

Note that if the path for $FUSEKI\_CONFIG contains a drive letter it should be written as file://{drive-letter}:/{rest-of-path}. The path names in this command should use forward ‘/’ characters.

Verify fuseki is running. There should be a directory ‘tdb’ in $FUSEKI\_DATABASES/$DATASET. A browser accessing <http://$SERVER:3030/> should see the Fuseki welcome page. Errors may be reported on the console or in $FUSEKI\_LOG/fuseki.log.

It is best to run Fuseki as a Windows service so that it can be configured to start automatically when the server is started and to be restarted if it stops unexpectedly. This can be done using the SRVANY tool and manually editing the registry. There are other tools that this easier to do such as http://nssm.cc/ , but these may not be authorized for use and do run with administrator privileges.

The appropriate mechanism to use to set Fuseki up as a service is beyond the scope of this document.

## Installing Tomcat

Tomcat is a web application container that will be used to host the Elda application. We suggest installing Tomcat version 7. Follow the instructions at <http://tomcat.apache.org/tomcat-7.0-doc/setup.html>. The directory in which Tomcat is installed will be referred to as $TOMCAT\_HOME.

Retrieve the page <http://$SERVER:8080/> with a browser and the Tomcat welcome page should be displayed.

## Installing a Web Server

These instructions install Apache. An alternative Web server application with an equivalent configuration may be substituted. There are many ways to configure the Web server appropriately. These instructions may be amended to suit.

Follow the instructions at [http://httpd.apache.org/docs/2.2/platform/windows.html to install Apache 2.2](http://httpd.apache.org/docs/2.2/platform/windows.html%20to%20install%20Apache%202.2). The home directory in which Apache is installed will be referred to as $APACHE\_HOME. This defaults to C:\Program Files\Apache Software\Apache2.2.

Decide where the Apache content, log, cache and cache log files should go. These will be referred to as $APACHE\_ROOT, $APACHE\_LOG, $APACHE\_CACHE and $APACHE\_LOCK respectively. Possible values for these are $APACHE\_HOME/www, $APACHE\_HOME/logs, $APACHE\_HOME/cache and $APACHE\_HOME/tmp.

Ensure these directories exist and that the user the Apache server is running as has both read and write access to them.

First set up the virtual host for data.sedgemoor.gov.uk.

1. Create a directory $APACHE\_HOME/conf/vhosts
2. Copy the supplied apache-config/vhosts/data.sedgemoor.gov.uk.conf to $APACHE\_HOME/conf/vhosts
3. Edit $APACHE\_HOME/conf/vhosts to replace any variables with their actual values

Now configure the Apache server to activate the required modules and the virtual host:

1. There is a section in the file $APACHE\_HOME/conf/httpd.conf that controls the enabling of server modules. Remove the ‘#’ characters from the beginning of the lines referring to the following modules to enable them:
   1. cache\_module
   2. disk\_cache\_module
   3. expires\_module
   4. proxy\_module
   5. proxy\_ajp\_module
   6. proxy\_http\_module
   7. reqtimeout\_module
   8. rewrite\_module
2. After the line:  
    #Include conf/extra/httpd-vhosts.conf  
   add the line  
    Include conf/vhosts/data.sedgemoor.gov.uk.conf

Install the static content to be served by the Web server.

1. Copy the contents of the supplied directory apache-data to $APACHE\_ROOT
2. Restart the Apache service

Retrieve the page <http://$SERVER/> and a minimal welcome page should be displayed.

Retrieve the page <http://$SERVER/qonsole.html> and a form to query the database should be displayed. Click on the button to execute the query. No results will be returned as there is no data, but there should be no errors reported.

## Install Elda

Elda comes in two parts. Elda assets are the icons, stylesheets and other components that can be modified to change the appearance of Elda. Elda common is the Elda application itself.

To install Elda assets:

1. Download the Elda assets WAR file from <http://repository.epimorphics.com/com/epimorphics/lda/elda-assets/1.3.0/elda-assets-1.3.0.war>. A later version may be substituted if one is available.
2. Extract the contents to of the archive to $APACHE\_ROOT/data.sedgemoor.gov.uk/lda-assets. DO NOT overwrite any existing files installed from the package. WAR files are archive files and can be read by the ZIP program. The content of $APACHE\_ROOT/data.sedgemoor.gov.uk /elda-assets should include directories css, images, openlayers, scripts, velocity and xslt.

To install the Elda configuration:

1. Create the directory /etc/elda in the root of the ‘current drive’ of the Tomcat process.
2. Copy the supplied directory elda-config/conf.d into /etc/elda
3. Edit the file /etc/elda/conf.d/elda/cim.ttl to replace variable names with their values.

To install Elda:

1. Download the Elda common WAR file from <http://repository.epimorphics.com/com/epimorphics/lda/elda-common/1.3.0/elda-common-1.3.0.war>. Use the same version as used for Elda assets.
2. Copy or move this file to $TOMCAT\_HOME/webapps/elda.war. After a few seconds a directory named elda should appear in $TOMCAT\_HOME/webapps

Verify that Elda is present and running by accessing the page <http://$SERVER/api-config> which should display the Elda API configuration page.

## Debugging

Sadly, the manual installation described here is error prone. A single typo, even a difference in case, can cause the installation to fail in non-obvious ways.

Looking at the log files in $APACHE\_LOG, $TOMCAT\_HOME/logs and $FUSEKI\_LOG can give clues. Apache also generates events that can be viewed in the Windows event viewer.

# Installing the client Software

The client software uses both Java and Ruby programming languages. Both need to be installed on the client machine.

## Installing Java

Java is needed for the data conversion software.

If Java is not already installed, follow the instructions in 2.1 to install Java.

## Installing Ruby

Ruby is a programming language. It is needed to run the program that uploads data to the server.

If Ruby is not already installed, select an installer from <http://rubyinstaller.org/downloads/>. Pick the recommended version. At the time of writing this was Ruby 1.9.3-p551.

Download the installer and run it. You will see something like:

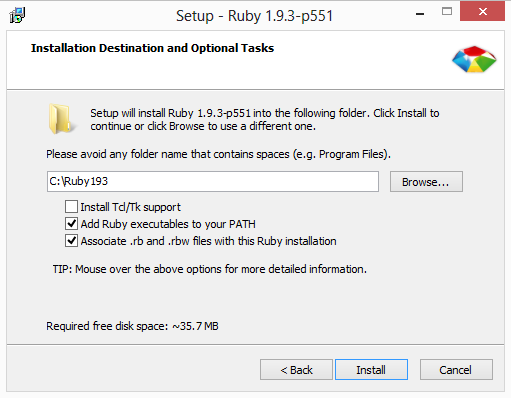


Figure 1 Ruby Installer

Tick the boxes to add Ruby executables to your PATH and associate .rb and .rbw files with this Ruby installation and click Install. After the files are installed click “Finish”.

Open a command prompt and run ruby –v. Ruby should report its version.

## Installing the Data Package and Scripts

Choose a directory where the data and scripts are to be installed. This is referred to as $DATA.

From the install package, copy the contents of the client directory into $DATA. After this is done, there should be a file $DATA\hinkleyC.xlsm. If necessary, rearrange the directory hierarchy so that hinklyC.xlsm is in $DATA. hinkleyC.xlsm is the spreadsheet that holds the master data. If a more recent version of the spreadsheet is available with more up to date data, then replace this file with the more recent version.

# Converting and Uploading Data

The data to be published is entered into the spreadsheet. This data needs to be converted to RDF and uploaded to the server. There are three steps to this process:

1. Export the data from the spreadsheet to CSV format files
2. Convert the CSV format files to RDF
3. Upload the RDF files to the server

## Export the spreadsheet data to CSV format files

To export the data in the spreadsheet, open the spreadsheet and ensure that macros are enabled. Switch to the Overview tab (the first one) and press the button labelled “Export All to CSV”.

A directory called ‘csv’ should be created next to the spreadsheet file and it should contain a number of csv files, e.g. agent.csv, community.csv etc.

When the spreadsheet is updated, these files can be updated by pressing the “Export All to CSV” button again. The data from individual sheets can be updated by pressing the “Export to CSV” button on that sheet.

## Convert the CSV format data to RDF format data

* Open a Windows CMD window
* cd to $BASE
* execute the command ‘convert’

The script should announce that is processing each of the CSV files in turn. For each file, the data converter will report the number of lines processed. Ignore any warning messages for now.

After executing the command, there should be a directory called ‘rdf’ in $BASE. This should contain a file with a “.ttl” extension for each of the CSV files in the csv directory.

## Upload the RDF format data to the server

* Open a Windows CMD window
* cd to $BASE
* Execute the command ‘upload’

The script should report uploading each of the RDF files individually.

Finally, the Elda cache should be cleared. This can be done by restarting Apache Tomat in the services management tool or by accessing the page <http://sedgemoor.data.gov.uk:8080/elda/control/show-cache> and clicking on the clear cache button.

The spreadsheet data has now been exported, converted to RDF and uploaded to the server.

## Verify the Data is Uploaded

Point a browser at http://$SERVER/id/community and something like the following screenshot should appear:

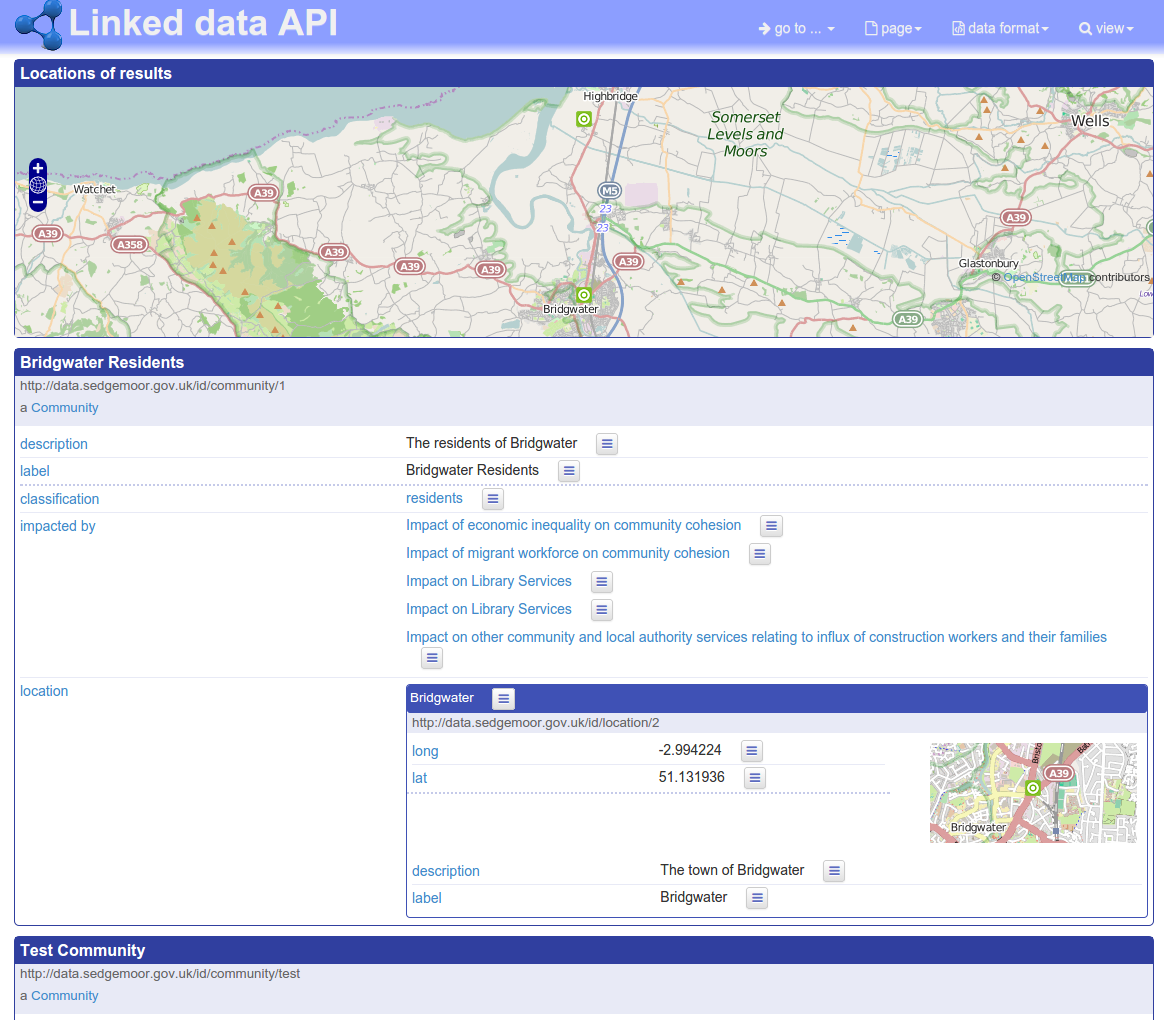


Figure 2 Elda Screenshot showing uploaded data

Navigate to other entry points (/id/impact and /id/project) and verify the expected data is present.