FUSION EDGE SERVER

Version 1.0.x

DEPLOYMENT INSTRUCTIONS AND USER GUIDE

Fusion Edge Server: 2.x
Document Last Updated: June 7th 2019

Version 20190607

1 Contents

Version History

Version #	Implemented By	Revision Date	Reason
20190607	Matt Nelson	07 June 2019	Added sender.id and api.version to properties file

2 Overview

Fusion Edge Server is a solution for supporting high performance data and metadata dissemination, with high throughput. The Fusion Edge Server serves all data, related metadata, including reference metadata, from its highly optimised in memory store and as such has no reliance on a database. The Fusion Edge Server has been designed to support the use case of high availability of data through load balancing.

Fusion Edge Server provides access to its stored data and metadata through SDMX compliant web services, and includes support for non-SDMX data formats such as CSV and Excel. The Fusion Edge Server extends the SDMX web services by providing additional information such as data metrics, and data search.

Fusion Edge Server is compatible with Fusion Data Browser and Fusion XL, providing an off-the-shelf solution for both web based data exploration, and from within Microsoft Excel. Connectivity to statistical software such as R and Matlab can be achieved using third party SDMX Connector libraries such as the SDMX Connectors project (https://github.com/amattioc/SDMX).

3 Deployment

3.1 Deployment Using Tomcat

Fusion Edge Server must be run within a Java Servlet Container. Metadata Technology recommends using Apache Tomcat as the Java Servlet Container.

Fusion Edge Server consists of a single .war file called *FusionEdgeServer.war*. This file needs to be copied into the directory: *<TOMCAT_HOME>/webapps* then the Tomcat server should be started. During the Tomcat start-up process, the war file will be unpacked into a directory of the same name, for example:

<TOMCAT_HOME>/webapps/FusionEdgeServer

Please check the Tomcat log files to ensure that Fusion Edge Server has deployed correctly. On successful launch the Fusion Edge Server product information service can be queried from the following URL:

http://[server]:[port]/FusionEdgeServer/ws/fusion/info/product

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/FusionEdgeServer

3.2 Memory Requirements

The Fusion Edge Server memory requirements depend on the volumes of data to be published. A minimum of 2Gb memory should be allocated. An extra 100Mb being allocated for every 2-3million observations should be sufficient.

3.3 Configuring Tomcat Memory

3.3.1 Tomcat Launched from File System

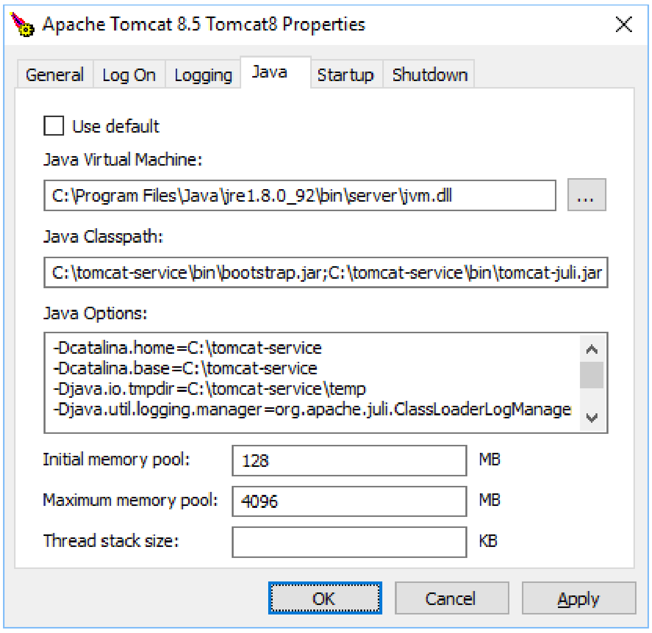
It is important to override the default Tomcat server memory settings as the default Tomcat settings will not be adequate to run the Fusion Edge Server. If Tomcat is running as a Windows service, please refer to the next section. When launching from the file system, overriding the default memory settings can be achieved by placing a *setenv.bat* (Windows) or *setenv.sh* (Unix) file into the Apache Tomcat *bin* folder. The setenv file is used to define environment variables for the JVM environment, and as such can be used to define more than just memory settings.

The recommended minimum setting is 2Gb Heap Memory, set using the JVM parameter –Xmx2G.

The Fusion Edge Server 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.

3.3.2 Tomcat Launched As A Service

When Tomcat is running as a service the Apache Configuration window must be used to configure memory settings. This window has a tab labelled “Java”. From this tab the appropriate settings can be applied.



☐ Use default

Java Virtual Machine:
C:\Program Files\Java\jre1.8.0_92\bin\server\jvm.dll

Java Classpath:
C:\tomcat-service\bin\bootstrap.jar;C:\tomcat-service\bin\tomcat-juli.jar

Java Options:

```
-Dcatalina.home=C:\tomcat-service
-Dcatalina.base=C:\tomcat-service
-Djava.io.tmpdir=C:\tomcat-service\temp
-Djava.util.logging.manager=org.apache.juli.ClassLoaderLogManage
```

Initial memory pool: 128 MB
Maximum memory pool: 4096 MB
Thread stack size: KB

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 Node 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.

3.4 Edge Server Directory

The Fusion Edge Server does not require a database, and as such makes use of the file system to read data and metadata files. The location of the Fusion Edge Server Directory defaults to:

```
<user home>/MetadataTechnology/FusionEdgeServer
```

On a Windows 10 Operating System this will typically be:

```
C:\users\<your user name>\MetadataTechnology\FusionEdgeServer
```

On a Unix Operating System, the likely location is:

```
/home/<your user name>/MetadataTechnology/FusionEdgeServer
```

The location of the Edge Server Directory can be changed by supplying a Java System variable to the JVM on tomcat launch.

The easiest way to achieve this is to create a new file named `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 directory which should be in the appropriate format for your operating system. To illustrate this:

```
SET JAVA_OPTS=-DEdgeServerDir=c:/dir/MyDirectory
```

(For Windows systems)

```
export JAVA_OPTS=-DEdgeServerDir =/dir/MyDirectory
```

(For Unix systems)

Note: Fusion Edge Server will NOT start if this value is incorrect or if this directory cannot be written to.

The Fusion Edge Server creates the following directories on start-up, if they do not already exist. The directories are:

- **sweep** – This folder is periodically polled by the Fusion Edge Server to check for new data
- **processing** – The Fusion Edge Server copied swept files into this folder for processing
- **processed** – The Fusion Edge Server moves successfully imported files into this folder, and imports any content from this folder on application start-up
- **tmp –** This folder is used to store both short lived temporary files, and longer lived cached datasets

3.5 Edge Server Properties Configuration

3.5.1 Properties File

The Fusion Edge Server provides configuration options on various aspects of the Edge Server. To set these configurations, create a file under the Edge Server Directory with the name: **edgeserver.properties**.

3.5.2 Edge Server Public URL

The Edge Server application and web service URL should be defined in the edge server properties file. These URLs are used in responses to structure queries which include links to associated reference metadata, and stub structures which link back to the full artefact.

The URLs are split into two parts, the application URL, which should be a reference to the public application server, and the rest url which is the POSTfix URL to the web service, for example:

```
app.url=https://myserver.myorg.org/Edge
rest.url=/sdmx
```

The above setting would make the SDMX web service entry point:
`https://myserver.myorg.org/Edge/sdmx`

3.5.3 Limiting Series and Observations

To place a limit on the number of series and the number of observations that can be queried via the web service the following two properties are supported:

```
limit.obs=12345
limit.series=1234
```

The numeric value indicates the maximum number of series or observations that can be returned from a single data query. If the limit is exceeded an HTTP status code of 413 (Payload Too Large) will be returned to the client.

3.5.4 Limiting Supported Data Formats

To enforce a restriction on which data formats are supported use the format.data property as follows:

format.data=[format1],[format2]

The supported formats are:

Format	String
SDMX Generic v1.0	application/vnd.sdmx.genericdata+xml;version=1.0
SDMX Generic v2.0	application/vnd.sdmx.genericdata+xml;version=2.0
SDMX Generic v2.1	application/vnd.sdmx.genericdata+xml;version=2.1
SDMX Generic v2.1	application/vnd.sdmx.generictimeseriesdata+xml;version=2.1
SDMX Structure Specific v1.0	application/vnd.sdmx.structurespecificdata+xml;version=1.0
SDMX Structure Specific v2.0	application/vnd.sdmx.structurespecificdata+xml;version=2.0
SDMX Structure Specific v2.1	application/vnd.sdmx.structurespecificdata+xml;version=2.1
SDMX Structure Specific v2.1	application/vnd.sdmx.structurespecifictimeseriesdata+xml;version=2.1
SDMX EDI	application/vnd.sdmx.edi
JSON	application/vnd.sdmx.json
JSON	application/vnd.sdmx.data+json
RDF Turtle	application/vnd.rdf+turtle
RDF Json	application/vnd.rdf+json
RDF XML	application/vnd.rdf+xml
FusionXL Plugin Format	application/vnd.fusionxl
Excel	application/vnd.openxmlformats-officedocument.spreadsheetml.sheet
CSV (default)	application/vnd.csv
CSV SDMX	application/vnd.csv-sdmx
CSV Time Series	application/vnd.csv-ts
CSV Custom Format	application/vnd.csv+,+:+true+true

Example:

format.data=application/vnd.sdmx.generictimeseriesdata+xml;version=2.1, application/vnd.csv

3.5.5 Limiting Supported Structure Formats

To enforce a restriction on which structure formats are supported use the format.structure property as follows:

format.structure=[format1],[format2]

The supported formats are:

Format	String
SDMX EDI	application/vnd.sdmx.structure;version=edi
SDMX v1.0	application/vnd.sdmx.structure+xml;version=1.0
SDMX v2.0	application/vnd.sdmx.structure+xml;version=2.0
SDMX v2.1	application/vnd.sdmx.structure+xml;version=2.1
Excel	application/vnd.openxmlformats-officedocument.spreadsheetml.sheet
JSON	application/json

Example:

format.structure= application/vnd.sdmx.structure;version=edi, application/json

3.5.6 CSV output format

To modify how the default CSV output is returned, the following properties are supported:

```
csv.delimiter=,
csv.labels=ID
csv.showSeries=true
```

The delimiter is the value of the character that will be output between the elements (defaults to comma). The values for 'Labels' must be one of the following: ID; NAME; ID_AND_NAME (defaults to ID_AND_NAME). 'Show Series' defines whether to output the series key in the output (defaults to true).

3.5.7 Sender Id

A Sender Id is put on the top of SDMX Datasets in the Header Section. This defaults to UNKNOWN, but can be set using the following property:

sender.id=YOUR_ID

The Sender Id must contain only Latin alpha-numeric characters, (A-Z, 0-9) and can also contain the following characters: **_**, **-**, **$**, **@**

Example: MN@123-ORG

4 Loading Edge Server Content

4.1 Overview

The Fusion Registry is used to generate the data and metadata content for the Fusion Edge Server. Please refer to the relevant Fusion Registry User Guide for more information. The output of the Fusion Registry is a single file called **node.zip**. This file should be copied or moved to the Fusion Edge Server's **sweep** directory.

The Fusion Edge Server will move the zip file into the **processing** directory, importing the structural metadata, reference metadata, and data content. The zip file is then moved into the **processed** directory and the structural information will be available from the Fusion Edge Server's web services.

The Fusion Edge Server treats each node.zip file as a full replace. The old environment will continue to be available via the web services until the full replacement environment is built. Once the replacement environment is ready, the data, structural metadata, and reference metadata services will all be switched to the new environment at exactly the same instance and the new service will be live. After the environment has been switched, the old environment will be discarded. If the Fusion Edge Server fails to read any part of the new environment, the publish process will fail and the old environment will continue to remain live. The Fusion Edge Server can continue to serve web service queries during the publication process, however the service will respond slower due to the increased CPU requirements of publication.

4.2 Embargo

It is possible to add an embargo date to the node.zip file. An embargo date enables the zip file contents to be read into the Fusion Edge Server to rebuild the environment, but the new environment will not be made available via the web services until the embargo time has passed.

To add an embargo to a node.zip file, rename the file to be node_[embargo time].zip where [embargo time] is UNIX Epoc Time (the number of milliseconds since 1970). For example to add an embargo to a node.zip file for the 29th January 2019 13.30am GMT, the node file would be named:

node_1548801000000.zip

The node.zip file (with embargo time) is placed into the sweep folder, and will be moved into the processing folder automatically by the Fusion Edge Server. The Fusion Edge Server will load the contents into memory an build a new environment, at this point in time the Fusion Edge Server will have two complete environments: the environment that is serving content to the web services; the embargo environment, which is built but has no services attached to it. When the embargo time passes, the embargo environment becomes the live environment, and the live environment is discarded. The node_embargo.zip file is automatically moved from the processing to the processed folder (renamed to node.zip). Should the Fusion Edge server be restarted when the embargo environment is still in the processing directory, it will be read back in, and the embargo time will be maintained.

It is possible to deploy multiple embargo environments, each will be read into memory and held until the embargo time has passed. It is worth noting that each node.zip contains a full replacement of the complete environment, so ensure there is enough memory allocated to hold multiple environments.

5 Web Services

The data held in the Fusion Edge Server is made available via SDMX compliant web services. The web service entry point for the Node is available at:

http(s)://[server]:[port]/FusionEdgeServer/ws/public/sdmxapi/rest

Full web service documentation is provided in a separate document.

6 Caching

The Fusion Edge Server provides multiple caching layers to help improve performance. The in-memory database maintains a local low footprint in-memory cache. In addition, each dataset which results from a data query is written to the **tmp** directory in the Fusion Edge Server Directory. Repeated queries will be served directory from the cached file, taking into account aspects which may change the data content such as response language (for datasets that include multilingual labels).

Client-side caching is supported through both the **If-None-Match**, and **If-Modified-Since** HTTP Request headers. Most web browsers will take advantage of the client-side caching with no additional configuration required.