FUSION NODE

Version 1.0.x

FUSION NODE – DATA DISSEMINATION VERSION 1

Node Overview

The Fusion Node Data Dissemination provides read only REST web services which are secure and highly optimised for data dissemination.

Fusion Node: 1.0.x
Document Last Updated: October 7th 2017

Version 20170907

1 Contents

2 Overview

The Fusion Node provides a standalone web application for data dissemination. A single Node instance can be run, or multiple instances can be load balanced over multiple servers, ensuring a highly available SDMX service.

The Node is fed information either from a Fusion Registry Node export, or from individual structure and data files, which can be transferred to the Node file system.

The Node generates a folder structure on the client system based on the SDMX structures the Node is provided with. To load data into the node, the data reporter simply places a SDMX, CSV, or Excel file into the relevant folder on the file system, the node validates the data, and loads it into memory for dissemination.

The Node runs a highly optimised data store (the Fusion Node Data Store) which provides fast access to data, exposed via SDMX compliant Web Services, and as such the Node can be used to serve websites as well as application to application communication.

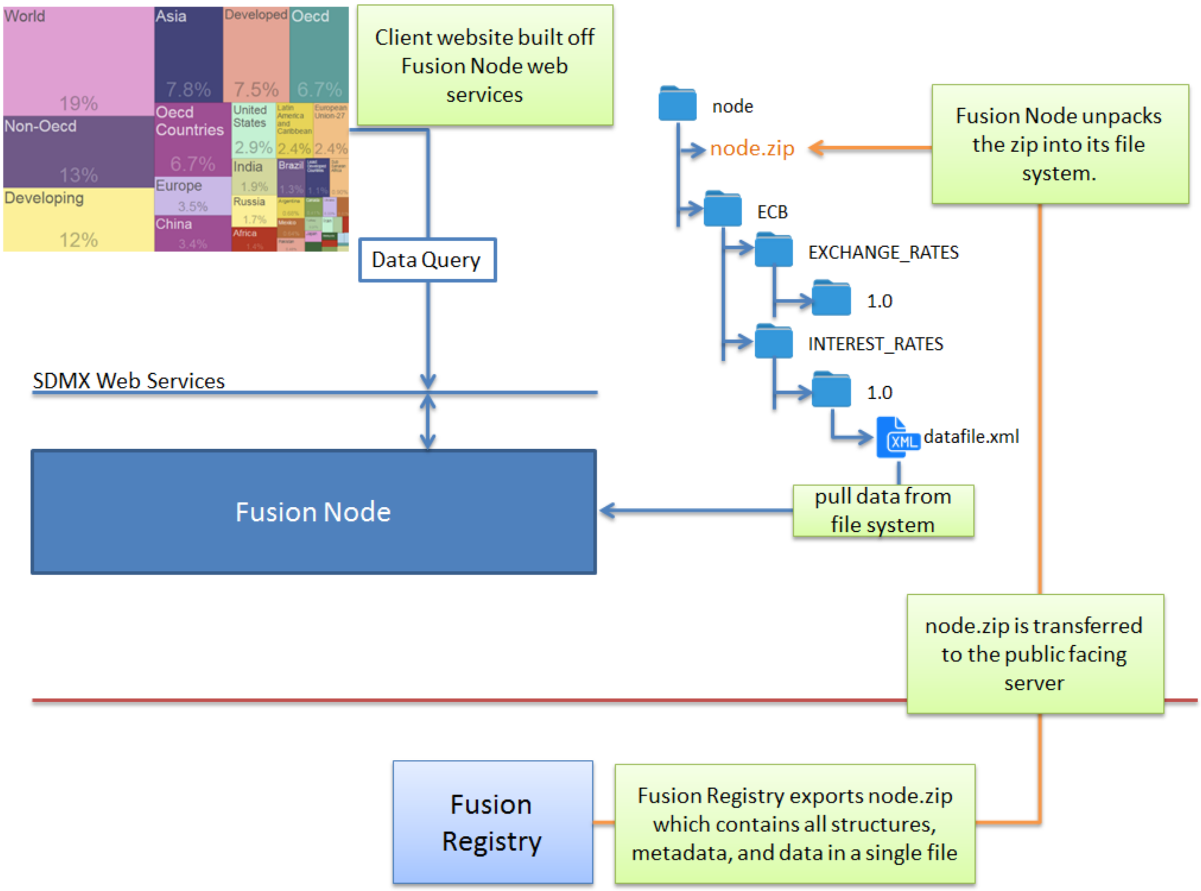


Figure 1 showing the node architecture

3 Deployment

3.1 Deployment Using Tomcat

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

Fusion Node consists of a single .war file called *FusionNode.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/ FusionNode

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

http://[server]:[port]/ FusionNode /

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/ FusionNode /

3.2 Memory Requirements

The node memory requirements depend on the volumes of data to be held on the node. 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 – 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 Node. Unless you are running Tomcat as a Windows service (see next section) 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:

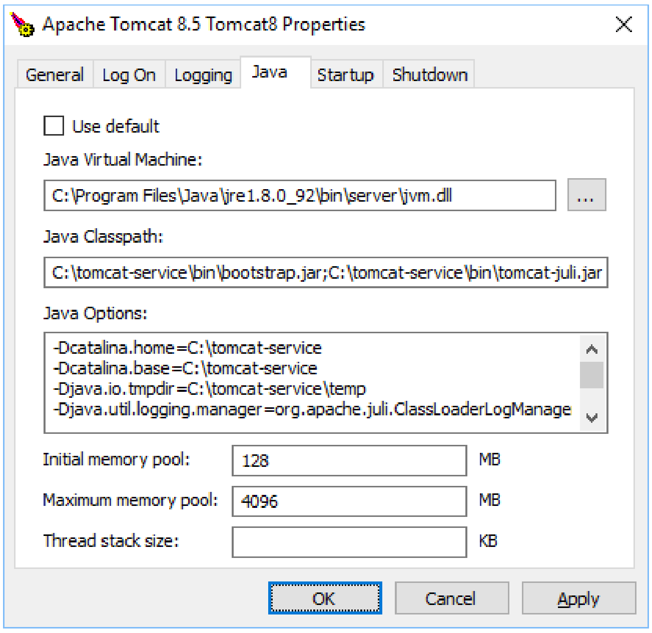
- **2Gb Heap Memory**
- **256M PermGen Space**

The Fusion Node 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.

Version 1.8 and later versions of Java removed the concept of PermGen Space. If you are using a Java runtime of version 1.8 and start the Fusion Node with the PermGen setting, a warning will be issued, but this will not stop the Node from starting successfully.

3.4 Configuring Tomcat Memory – From 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.

3.5 Node Properties and Licence File Location

Fusion Node stores all configuration settings in a properties file. In addition a licence file is required to start Fusion Node. If the licence file is missing, the User Interface will prompt for the license to be supplied, and will be written to the licence file location.

The location of the properties and licence file defaults to

```
<user home>/MetadataTechnology/FusionNode
```

On a Windows 10 Operating System this will typically be:

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

On a Unix Operating System, the likely location is:

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

The location of the properties and licence file 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=-DNodeProperties=c:/dir/MyDirectory
```

(For Windows systems)

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

(For Unix systems)

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

3.6 Fusion Node File System (FNFS)

Node obtains and stores all information on the hard disk under a location called the FNFS. The default location of the FNFS is:

```
<user home>/MetadataTechnology/FusionNode/FNFS
```

On a Windows 10 Operating System this will typically be:

```
C:\users\<your user name>\MetadataTechnology\FusionNode\FNFS
```

On a Unix Operating System, the likely location is:

```
/home/<your user name>/MetadataTechnology/FusionNode/FNFS
```

It is possible to change this location from the Fusion Node settings page documented in the next section of this document.

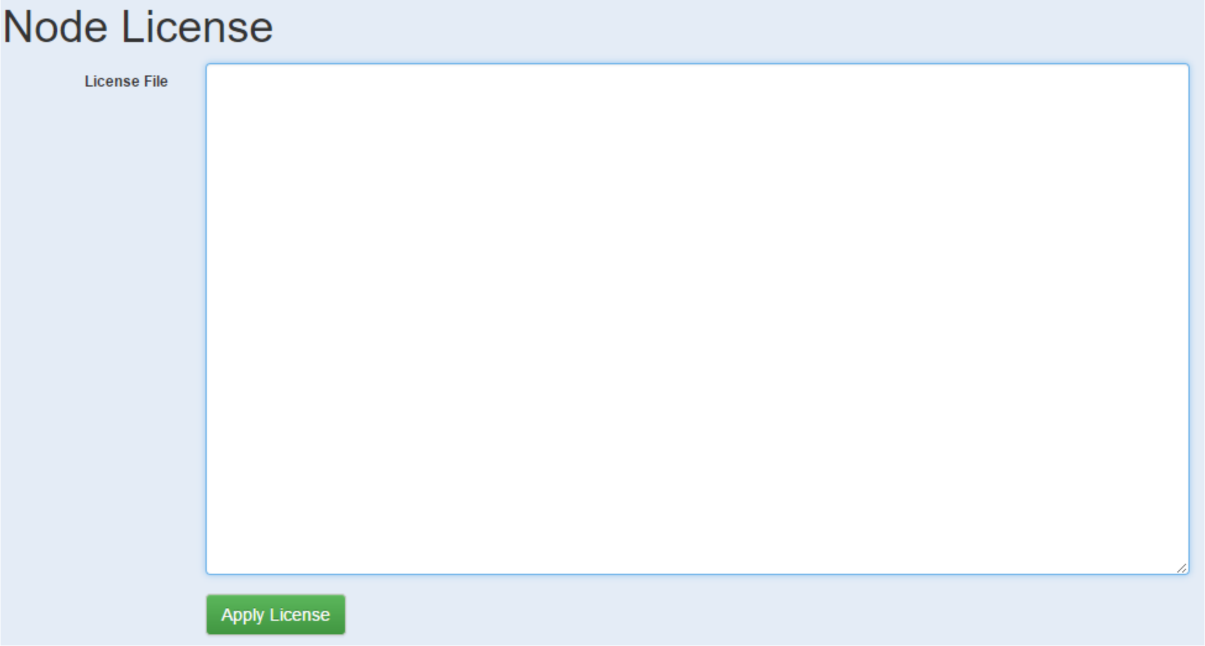
It is important that the Fusion Node has both read and write permissions for the FNFS and its subdirectories.

4 Configuring the Node

After launching Tomcat use the web browser to open the Node home page

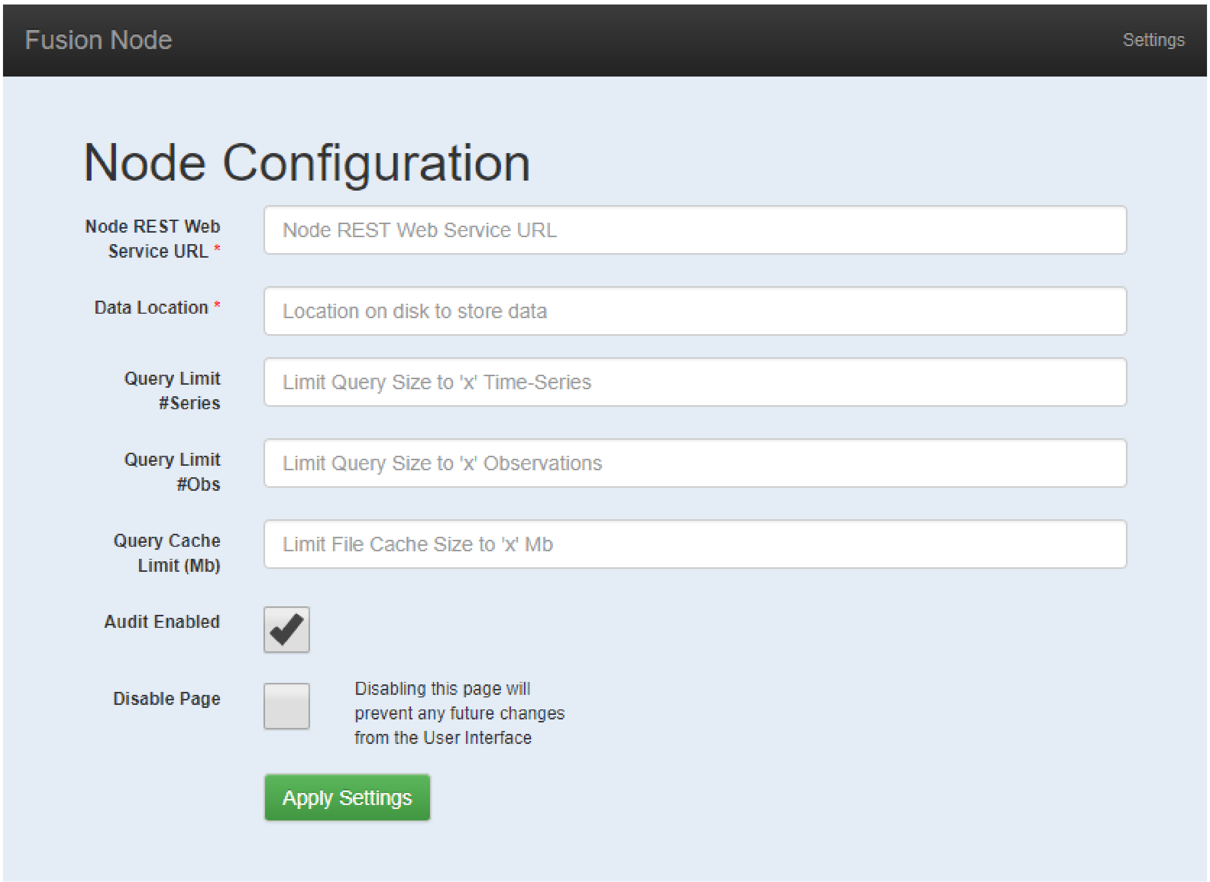
http(s)://[server]:[port]/ FusionNode

When the node is first launched, the Node License page will be displayed. The license file should be copied and pasted into the text area followed by clicking the Apply License button.



Node License

After successfully applying the license, the Node Configuration page will be displayed, node configuration must be completed before the node can be used.



Fusion Node

Settings

Node Configuration

Node REST Web Service URL * : Node REST Web Service URL

Data Location * : Location on disk to store data

Query Limit #Series : Limit Query Size to 'x' Time-Series

Query Limit #Obs : Limit Query Size to 'x' Observations

Query Cache Limit (Mb) : Limit File Cache Size to 'x' Mb

Audit Enabled : ☑

Disable Page : □ Disabling this page will prevent any future changes from the User Interface

Apply Settings

Figure 2 showing the node configuration page

The Node Configuration requires the following settings:

Setting	Definition
Node REST Web Service URL	This is the URL of the nodes public SDMX web service. This is required by the node for certain SDMX responses, for example applying Reference Metadata links into structure and data messages. The node web service URL is *http(s)://[server]:[port]/ FusionNode/ws/public/sdmxapi/rest* If using a Proxy server such as Apache (recommended) to route public traffic then this URL must be the public facing URL, as the Hub must be able to communicate with it.
Data Location	This is the location of the Fusion Node File System (FNFS). This folder is where the node.zip file is deployed to, and where structural metadata, reference metadata, and data live. Any cached and temporary data live under the tmp subfolder. Any audited information is stored in CSV files under the Audit subfolder.
Query Limit #Series	Optional setting. Enforcers a limit on the number of series that can be queried
Query Limit #Obs	Optional setting. Enforcers a limit on the number of observations that can be queried
Query Cache Limit (Mb)	Optional setting. If set, caching will be enabled, with the given limit.
Audit Enabled	If true, Fusion Node will audit queries to its REST web service URL. The

	Audit information is written out as CSV to the Audit sub folder of the FNFS. Audit files are automatically zipped when they read 10Mb in size. **Note:** Fusion Node does not automatically delete old audit files.
Disable Page	If checked, the settings page will no longer be available from the User Interface. To re-enable the settings page, the node properties file will need to be manually edited to remove this entry, and the Node server must be restarted for the modification to take effect.

Node settings can be redefined at any time by clicking the Settings button which is located in the header, at the top right, of the home page.

5 Loading Node Content

5.1 Node Folder Structure

The Fusion Node File System (FNFS) is configured in the settings page of Fusion Node.

The folder structure is based on the Provision Agreements the Node is supplied with. A Provision Agreement is an item of metadata which links a Data Provider to a Dataflow. A data reporter will always report data against the provision agreement, and from this the Hub is able to identify who the Data Provider is and which Dataflow they are reporting data for.

Folders are created for each provision agreement that the node's data reporter has Provision Agreements for, as defined in the Hub. The folder structure is as follows:

Provision Agreement Owning Agency Id → Provision Agreement Id → Provision Agreement Version

5.2 Loading from Fusion Registry

The Fusion Registry can export a single node.zip file, which can be provided to the Fusion Node by simply placing the file into the node folder.

The Fusion Node will unzip the contents and delete the node.zip file. The unpacked file will then be read back in by the Fusion Node and ready for dissemination.

This process is extremely fast, the node file system is scanned every 5 seconds, and it can take as little as 10 seconds to read in over 20 million observations, ready for dissemination via the web services.

5.3 Loading Structures

To load structures into the Fusion Node without Fusion Registry, simply place the structure file, which must be called 'structures', into the Node file system. The Fusion Node will read the file and create the node folder structure based on the Provision Agreements in the file. The structures file must be in SDMX format.

5.4 Loading Reference Metadata

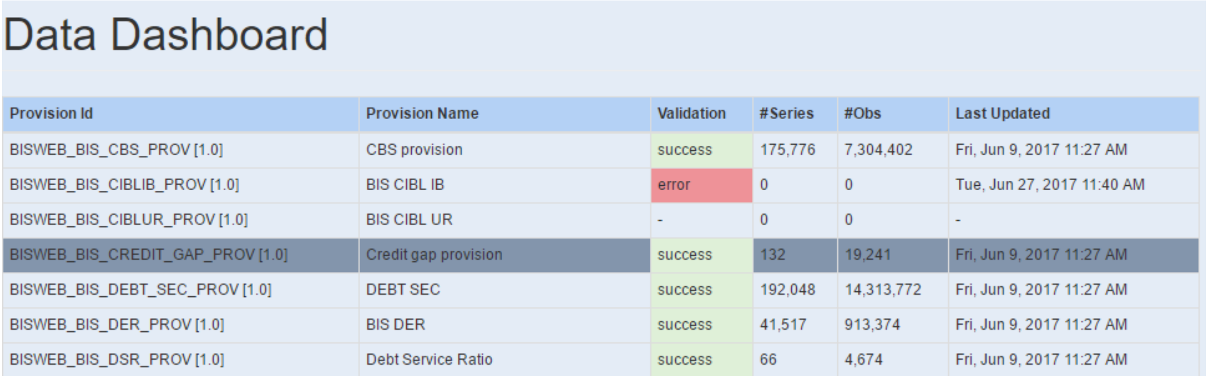
To load reference into the Fusion Node without Fusion Registry, simply place the metadata file, which must be called 'metadata', into the Node file system. The metadata file must be in SDMX format.

5.5 Loading Data

To load new data, copy the data file into the leaf folder (Provision Agreement Version). The file can be in any supported node format.

The node will detect the presence of a new file and read it for validation. If it passes the validation stage, it will be loaded into the node's data store, and the data will be available from the node's web service.

The Node dashboard updates every 5 seconds to reflect the current status of each Provision Agreement. If any data loads result in error, this will be displayed in the dashboard. The error cell can be clicked to open up the full error details.



Data Dashboard

Provision Id	Provision Name	Validation	#Series	#Obs	Last Updated
BISWEB_BIS_CBS_PROV [1.0]	CBS provision	success	175,776	7,304,402	Fri, Jun 9, 2017 11:27 AM
BISWEB_BIS_CIBLIB_PROV [1.0]	BIS CIBL IB	error	0	0	Tue, Jun 27, 2017 11:40 AM
BISWEB_BIS_CIBLUR_PROV [1.0]	BIS CIBL UR	-	0	0	-
BISWEB_BIS_CREDIT_GAP_PROV [1.0]	Credit gap provision	success	132	19,241	Fri, Jun 9, 2017 11:27 AM
BISWEB_BIS_DEBT_SEC_PROV [1.0]	DEBT SEC	success	192,048	14,313,772	Fri, Jun 9, 2017 11:27 AM
BISWEB_BIS_DER_PROV [1.0]	BIS DER	success	41,517	913,374	Fri, Jun 9, 2017 11:27 AM
BISWEB_BIS_DSR_PROV [1.0]	Debt Service Ratio	success	66	4,674	Fri, Jun 9, 2017 11:27 AM

5.6 Updating Data

To update data from the file system, place the new data file in the Provision Agreement folder. Delete the file named **fds**, this will force a re-read of the data file(s).

It is possible to have more than one data file for a particular Provision Agreement. The Node will read all data files in the provision agreement folder, and consolidate them into a single dataset. If there are no duplications and the files are valid, then all data sets will be loaded into the Node data store.

If the date modified date of the new data file is later than any other file in the directory, then the node will automatically revalidate all datafiles for import. Otherwise files marked as .fds and .err should be deleted to force a revalidation.

5.7 Deleting Data

Data can be deleted by deleting the file named **fds** which is located in the Provision Agreement folder. If there are any data files, these should be deleted too, otherwise they will be loaded back into the Node automatically, and the fds file will be recreated.

6 Data view and web services

6.1 SDMX Web Services

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

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

Node web service documentation is provided separately

6.2 Excel Export

Data can be exported from the node in Excel format, note for larger datasets this may be an expensive operation. To export data in excel, use the node web service to select the provision agreement, and then click the ‘Export Excel’ button.

6.3 Building a Custom Web User Interface

In addition the IMJS JavaScript library can be used to build upon the node web services in order to build a web user interface for data exploring, visualisation, and export.

IMJS documentation is available on the Metadata Technology website

http://www.metadatatechnology.com/FusionRegistry9/IMJS/

6.4 Performance

In terms of performance, the node web services can support over 200 requests per second, depending on the size of data query. The node supports query piggybacking, which prevents the same query running concurrently multiple times on the node. In addition the node provides a file system caching solution to ensure peak performance is met.

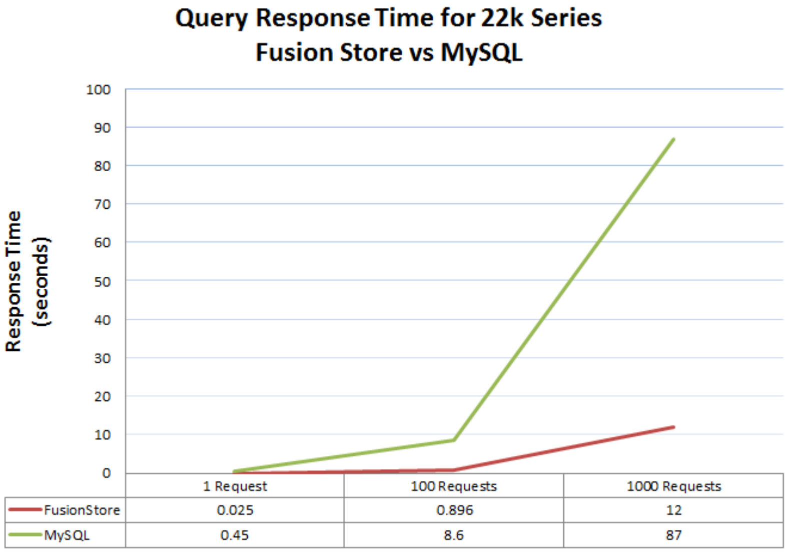
Benchmarking the node’s internal database (Fusion Store) against a relational database using an optimised star schema design, for the same dataset, the node provides a much improved service, even with all caching disabled.

Dataset Size:	500Mb SDMX file with 2.1 Million Observations
Tomcat Max Memory:	2Gb
Data Load:	58 Seconds

Query 22k Observations:

Num Requests	Concurrency	Time (s)	Requests Per Second
1	1	0.025	40
100	10	0.872	114
100	50	0.896	111
1000	50	12	82

When benchmarked against the same query for a MySQL instance (on a desktop PC) the graph looks as follows. **Note:** All caching is disabled:



	1 Request	100 Requests	1000 Requests
🟥 FusionStore	0.025	0.896	12
🟩 MySQL	0.45	8.6	87