

# FUSION REPORTING NODE

Version 2.0

## FUSION REPORTING NODE

### Node Overview

The Fusion Reporting Node is run locally by data providers to report data to a central HUB. The Node exposes SDMX compliant web services for the HUB to obtain data on demand. The Node can connect to a local database to load data into the Node on demand.

# Contents

1	Introduction .....	3
1.1	Scope of this document .....	3
1.2	Overview .....	3
2	Deployment.....	4
2.1	Fusion Reporting Node Distribution .....	4
2.2	Java (required) .....	4
2.3	Servlet Container (required) .....	4
2.4	Database (optional).....	4
2.5	Deployment Using Tomcat.....	4
2.6	Memory Requirements .....	4
2.7	Configuring Tomcat Memory – From File System .....	5
2.8	Configuring Tomcat Memory – From Service .....	5
2.9	Node Properties and Licence File Location.....	5
2.10	Fusion Node File System (FNFS).....	6
3	Licensing.....	7
4	Configuring Reporting Node .....	8
4.1	Server Configuration .....	8
4.2	Security Setup .....	9
4.3	Email Server .....	10
5	Data Dashboard .....	11
5.1	Overview .....	11
6	Loading Data .....	13
6.1	Loading Data from File .....	13
6.2	Loading Data from Database .....	13
6.2.1	Overview .....	13
6.2.2	Table or View Naming Convention .....	13
6.2.3	Column Naming Convention .....	13
7	Data Reporting Errors .....	15
8	Data view and web services.....	16
8.1	SDMX Web Services .....	16
8.2	Excel Export.....	16
8.3	Building a Custom Web User Interface .....	16
8.4	Performance .....	16
9	Resync structures with Hub .....	18



# 1 Introduction

## 1.1 Scope of this document

The purpose of this document is to provide an overview of how to install, configure and use Fusion Reporting Node in a Java Servlet Container.

## 1.2 Overview

The Fusion Reporting Node provides a standalone web application for the purpose of automatic data validation, transformation, and reporting to a central Fusion Registry server which acts as a Hub.

The Fusion Reporting Node obtains metadata from the Hub relevant to the data reporter that is running the Node. The metadata includes information about datasets that the data reporter has been configured to report data for and the related metadata which the Fusion Reporting Node uses to run its data validation service.

Data can be loaded directly into the Fusion Reporting Node in both Excel, and SDMX format. The data reporter can optionally link the Fusion Reporting Node to a MySQL, SQL Server, or Oracle database which can be used as a source of data.

The Fusion Reporting Node runs a highly optimised data store which provides fast access to data, exposed via SDMX compliant web services, and as such the Fusion Reporting Node can be used to provide data to a Registry hub as well as support other data services built on the Node's web services.

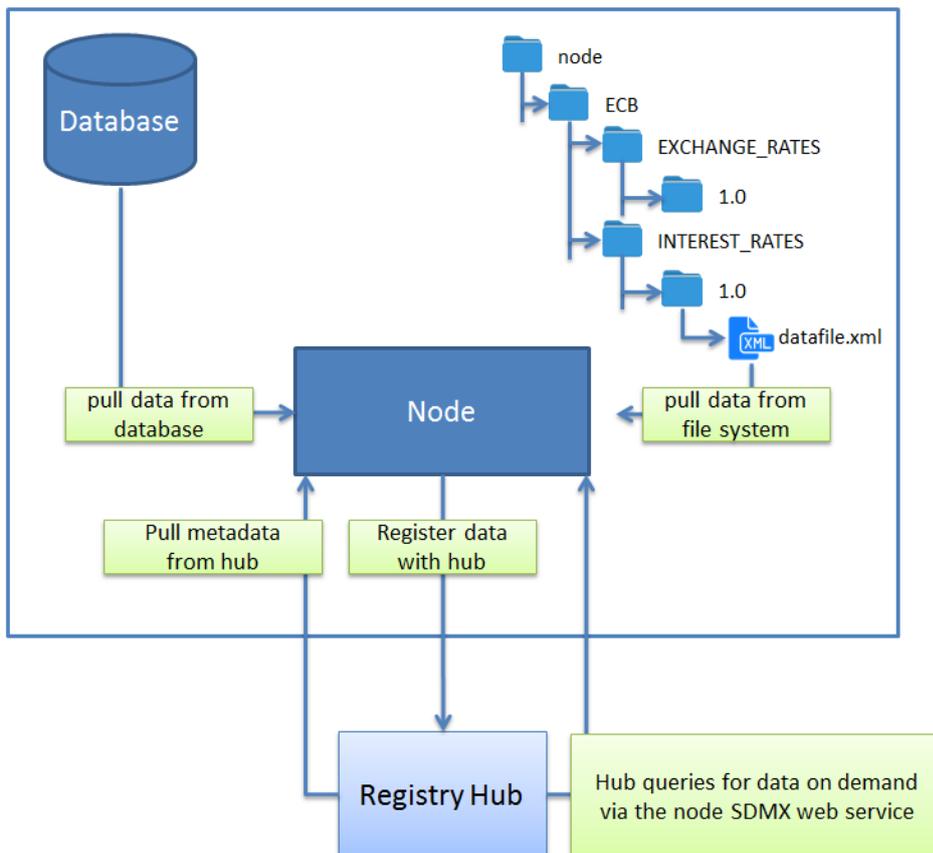


Figure 1 showing the Node architecture

## 2 Deployment

### 2.1 Fusion Reporting Node Distribution

Fusion Reporting Node is distributed as a single Web Application Archive (war) file: *FusionReportingNode.war* it should be deployed to a servlet container such as Apache Tomcat (a free product).

### 2.2 Java (required)

A Java Runtime Environment (JRE) version 1.8 or higher.

### 2.3 Servlet Container (required)

Fusion Reporting Node must be run within a Java Servlet Container. Metadata Technology recommends using Apache Tomcat as the Java Servlet Container, as this has been used during the testing lifecycle of Fusion Reporting Node. Fusion Reporting Node has only been tested in Apache Tomcat and therefore we cannot guarantee that Fusion Reporting Node will work with other Java Servlet Containers. Apache Tomcat 8 or above is supported.

The rest of this document will only give servlet container information regarding Apache Tomcat.

### 2.4 Database (optional)

Fusion Reporting Node may communicate with any SQL-92 compliant database. This distribution has only been set up to connect to MySQL, Oracle, and SQL Server databases, the minimum tested version for these databases are: MySQL 5.6, Oracle 10g and SQL Server 2010.

If your database is not one of these types, please contact Metadata Technology, as we may be able to add your database to the list of supported database management systems.

### 2.5 Deployment Using Tomcat

Fusion Reporting Node consists of a single .war file called *FusionReportingNode.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 Reporting Node war file will be unpacked into the directory:

```
<TOMCAT_HOME>/webapps/FusionReportingNode
```

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

```
http://[server]:[port]/FusionReportingNode
```

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/FusionReportingNode
```

### 2.6 Memory Requirements

The Fusion Reporting 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.

## 2.7 Configuring Tomcat Memory – From File System

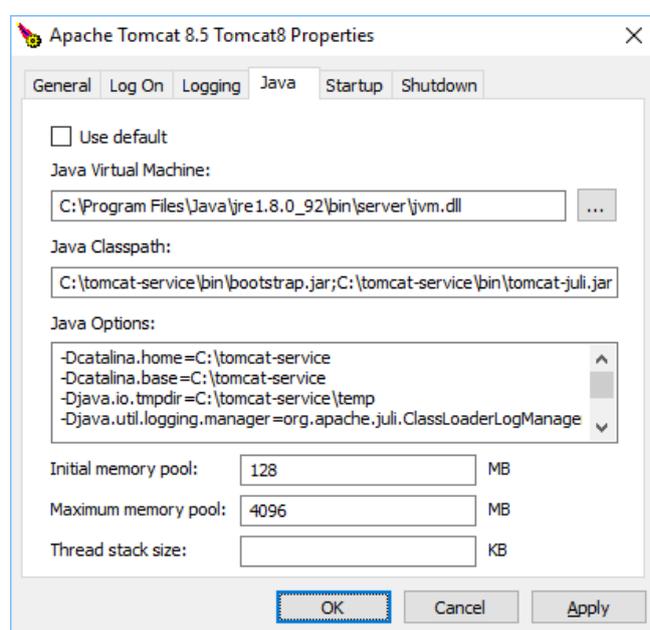
It is important to override the default Tomcat server memory settings as the default Tomcat settings will not be adequate to run the Reporting Node. Unless you are running Tomcat as a Windows service (see next section) overriding the memory settings can be achieved by placing a *setenv.bat* (Windows) or *setenv.sh* (Unix) file into the Apache Tomcat *bin* folder. The recommended minimum settings are:

- **2Gb Heap Memory**

The Reporting 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.

## 2.8 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.



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 Reporting 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.

## 2.9 Node Properties and Licence File Location

Fusion Reporting Node stores all configuration settings in a properties file. In addition, a licence file is required. If the licence file is missing, the User Interface will prompt for the licence 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/FusionReportingNode
```

On a Windows 10 Operating System this will typically be:

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

On a Unix Operating System, the likely location is:

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

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 tomcat's `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 Reporting Node will NOT start if this value is incorrect or if this directory cannot be written to.

## 2.10 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/FusionReportingNode/FNFS
```

On a Windows 10 Operating System this will typically be:

```
C:\users\\MetadataTechnology\FusionReportingNode\FNFS
```

On a Unix Operating System, the likely location is:

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

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

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

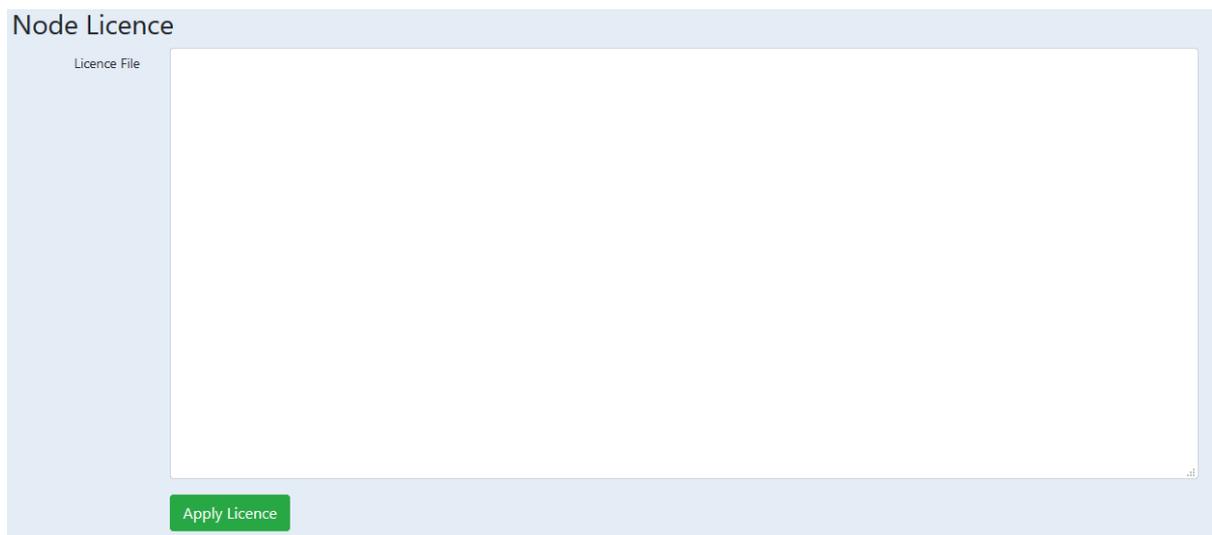
### 3 Licensing

After launching Tomcat use a web browser to navigate to the Reporting Node home page:

*http(s)://[server]:[port]/FusionReportingNode*

When the Reporting Node is first launched, the licence page will be displayed. The Reporting Node license is generated by the Fusion Registry administrator and contains important information such as the Fusion Registry location (URL) and the Data Provider that the license file was generated for. On applying the license file, the Fusion Reporting Node will use the license details to query the Fusion Registry for its content, and then apply filters on the content based on the Data Provider details. The result is a local copy of the Fusion Registry content, which is relevant to the Data Provider.

To apply the license file, open the file in a text editor, and copy the file contents into the text area in the web page of Fusion Reporting Node. Then click the “Apply Licence” button to confirm the licence.



Node Licence

Licence File

Apply Licence

*Figure 2 showing the Node licence page*

## 4 Configuring Reporting Node

### 4.1 Server Configuration

After successfully applying the licence, the Reporting Node Server Settings Configuration page will be displayed. The configuration details must be completed before the Reporting Node can be used.

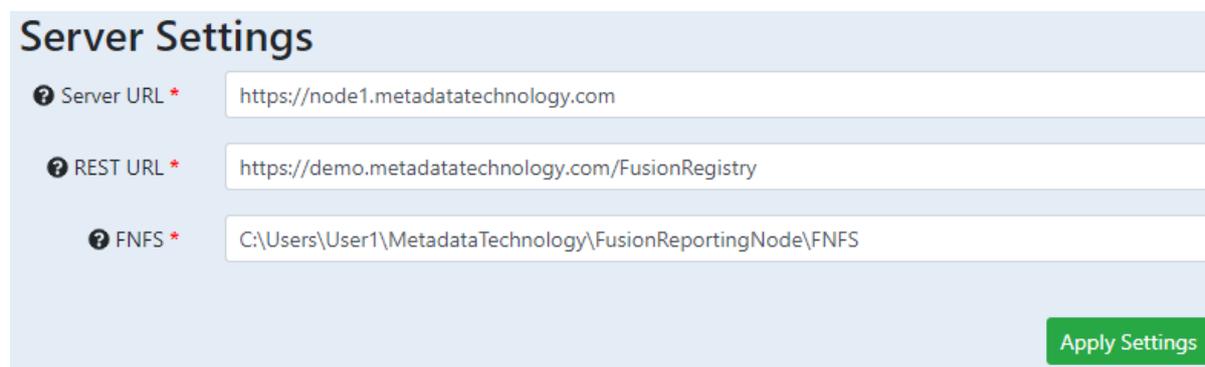


Figure 3 showing the Reporting Node configuration page

The configuration requires the following settings.

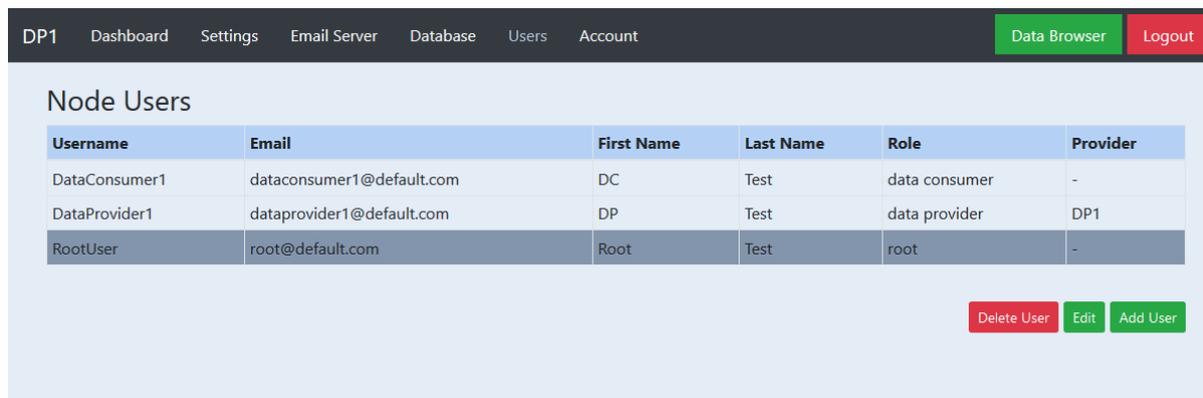
Setting	Definition
<b>Server URL</b>	This is the URL that the Reporting Node is installed on.
<b>REST URL</b>	<p>This is the URL of the Reporting Node's public SDMX web service. This is required by the Reporting Node to report to the Hub on data registration and must resolve to the web service of the Reporting Node.</p> <p>The Reporting Node web service URL is <i>http(s)://[server]:[port]/FusionNode/ws/public/sdmxapi/rest</i> If using a Proxy server such as Apache to route public traffic then this URL must be the public facing URL, as the Hub must be able to access URL to be able to perform data queries against it.</p>
<b>FNFS</b>	<p>This is the location of the Fusion Node File System (FNFS).</p> <p>This is where the folders are created for each Provision Agreement, which can be used to publish data to the Hub (drag and drop data file) and if using a database, where the pulled data is stored.</p> <p>In addition, any cached temporary data will be put into the temporary directory in the FNFS.</p> <p>Any audited information is stored in CSV files under the Audit subfolder.</p>

On applying these settings, the Fusion Reporting Node will obtain the structural metadata from the Fusion Registry hub server. This may take some time. On completion the Fusion Reporting Node Dashboard will be shown.

The Reporting Node settings can be modified at any time by clicking the 'Settings' control, in the top menu bar.

## 4.2 Security Setup

The Fusion Reporting Node is public until users are added. To add a user, click on the Users link in the header of the page. The first user added is the root user. Once the root user has been added further user types can be created.



Username	Email	First Name	Last Name	Role	Provider
DataConsumer1	dataconsumer1@default.com	DC	Test	data consumer	-
DataProvider1	dataprovder1@default.com	DP	Test	data provider	DP1
RootUser	root@default.com	Root	Test	root	-

Figure 4 showing the Users configuration page

Each user must be assigned one of the following roles:

- **Admin** – an admin user can perform any action
- **Data Manager** – can load and publish datasets, cannot change Node settings
- **Data Provider** – can load datasets into staging, cannot publish or remove published datasets, cannot change settings
- **Data Consumer** – can only view data

### 4.3 Email Server

An email server can be set up so that the Fusion Reporting Node can send out email notifications, including:

- Password reset links
- Data staged / published notifications
- Data registration error notifications
- Advanced release calendar notifications

To set up an email server click on the Email button in the header of page, and fill in the required details.

The screenshot shows the 'Email Server' configuration page. The navigation bar includes 'DP1', 'Dashboard', 'Settings', 'Email Server', 'Database', 'Users', and 'Account'. The 'Email Server' section contains the following fields and controls:

- SMTP Server:
- Port:
- Mail From Address (optional):
- Has Security:
- Username:
- Password:

At the bottom right, there are three buttons: 'Remove Email Server', 'Test Email', and 'Apply Settings'.

Figure 5 showing the Email Server configuration page

## 5 Data Dashboard

### 5.1 Overview

The screenshot shows the Fusion Reporting Node Dashboard. At the top, there is a navigation bar with links for ESTAT, Dashboard, Settings, Email Server, Database, Users, and a green Data Browser button. The main content area is titled "Data Dashboard" and features a table of provision agreements. The table has columns for Provision Id, Provision Name, #Series, and Last Updated. Below the table are buttons for View, Export, Describe, Data Entry, Load Dataset, and Resync Structures. The bottom section is split into "Staged Datasets" and "Published Datasets", each with a table showing File, Verified/Published, and Status, and buttons for View, Export, Re-Verify/Publish, and Delete.

Provision Id	Provision Name	#Series	Last Updated
ESTAT:CENSUS_ESTAT_HC42(1.0)	HC42 Provision	-	-
ESTAT:HC01(1.0)	HC01 Provision	-	-
ESTAT:HC02(1.0)	HC02 Provision	-	-
ESTAT:HC03(1.0)	HC03 Provision	-	-
ESTAT:HC04(1.0)	HC04 Provision	-	-
ESTAT:HC05(1.0)	HC05 Provision	-	-
ESTAT:HC06(1.0)	HC06 Provision	-	-
ESTAT:HC07(1.0)	HC07 Provision	-	-
ESTAT:HC08(1.0)	HC08 Provision	-	-
ESTAT:HC09(1.0)	HC09 Provision	-	-

Last Fusion Registry - Demo Server Sync: a few seconds ago

View Export Describe Data Entry Load Dataset Resync Structures

#### Staged Datasets

File	Verified	Status
No Files		

View Export Re-Verify Publish Delete

#### Published Datasets

File	Published	Status
No Files		

View Export Stage Delete

Figure 6 showing the Fusion Reporting Node Dashboard

The Dashboard of Fusion Reporting Node is split into three parts:

1. The Data Dashboard, showing all of the provision agreements that the Reporting Node is aware of as well as a series count of all data loaded in and the time it was last updated.
2. The Staged Datasets area, showing the status of dataset,s for the selected Provision Agreement, that have been loaded into the Reporting Node but have not yet been pushed to the Hub.
3. The Published Dataset area, showing the status of datasets for the selected Provision Agreement, that have been published to the Hub.

From the Data Dashboard, for the selected Provision Agreement, any published data can be viewed by clicking the View button. The data can be exported, by using the Export button. The Provision Agreement can be described in terms of Codelists and DSD. Clicking the “Describe” button will create an Excel spreadsheet with a number of worksheets. Each of these describes a single Codelist or the DSD. This can be helpful for creating valid data.

The button “Data Entry” when clicked prompts the user to specify the Dates for which data should be reported as well as the frequency (e.g. Annual, Monthly, etc.). Clicking “Create Worksheet” will create an Excel Spreadsheet which can be completed to provide data to the Reporting Node.

Once Data is in the Staged area it can be viewed, exported, etc. or promoted to the Published Datasets section. Data that has been published can be moved back to the Staged area at any time.

## 6 Loading Data

### 6.1 Loading Data from File

Data may be loaded into Reporting Node directly from a file. Selecting the appropriate provision agreement in the Data Dashboard and then clicking “Load Dataset” will display a modal prompting the user to specify which file to load. Data can be loaded in either SDMX or Excel format.

### 6.2 Loading Data from Database

#### 6.2.1 Overview

If a database connection has been defined, it will be possible to pull data from the database for the selected provision agreement. The ‘Database Sync’ button on the user interface, deletes all existing data in the Node, for the provision agreement, and pulls the data from the database into the Node’s folder system for import.

#### 6.2.2 Table or View Naming Convention

The database table or view must follow the naming convention of:

*[Provision Agency Id]\_[Provision Id]\_[Provision Version]*

The table name must be in upper case, and any ‘.’ or ‘-’ characters must be replaced with underscores ‘\_’. For example:

Provision Agency	Provision Id	Provision Version	Expected Table/View Name
<b>WB</b>	POVERTY	1.0	WB_POVERTY_1_0
<b>ONS.DEPT1</b>	GDP-DATA	2.1.1	ONS_DEPT1_GDP_DATA_2_1_1

The Provision Agreement agency, id, and version can be determined by looking at the folder structure in the Node, or by visiting the Hub Fusion Registry.

#### 6.2.3 Column Naming Convention

The database table or view is expected to contain a column for each of the Components of the Data Structure Definition that the Provision Agreement conforms to. For example:

DSD Component ID	Component Type	Column Id
<b>FREQ</b>	Dimension	FREQ
<b>REF_AREA</b>	Dimension	REF_AREA
<b>SERIES</b>	Dimension	SERIES
<b>Series_Title</b>	Series Attribute	SERIES_TITLE
<b>OBS_VALUE</b>	Observation Attribute	OBS_VALUE
<b>OBS_CONF</b>	Observation Attribute	OBS_CONF
<b>OBS_VALUE</b>	Observation Value	OBS_VALUE
<b>TIME_PERIOD</b>	Time Dimension	TIME_PERIOD

The value in the TIME\_PERIOD column must represent the same date, formatted according to the frequency. The following table shows an example of the observation frequency vs the expected date format.

Frequency	Format	Example
<b>Annual</b>	YYYY	2010
<b>Daily</b>	YYYY-MM-DD	2010-01-01
<b>Date Time</b>	YYYY-MM-DD-Thh:mm:ss	2010-01T20:22:00

<b>Monthly</b>	YYYY-MM	2010-01
<b>Quarterly</b>	YYYY-Qn	2010-Q1
<b>Semester</b>	YYYY-Sn	2010-S1
<b>Trimester</b>	YYYY-Tn	2010-T1

## 7 Data Reporting Errors

A dataset can result in error if either the database pull/data validation fails on the Node, or if the data registration process failed. In both cases, clicking on the table cell with label 'error' will display the error reason. If the data validation failed, the data can be deleted on the Node. If data registration failed, check the Node settings correctly specify the Node data web service, and then re-register.

## 8 Data view and web services

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

Node web service documentation is provided separately

### 8.2 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.

### 8.3 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/>

### 8.4 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

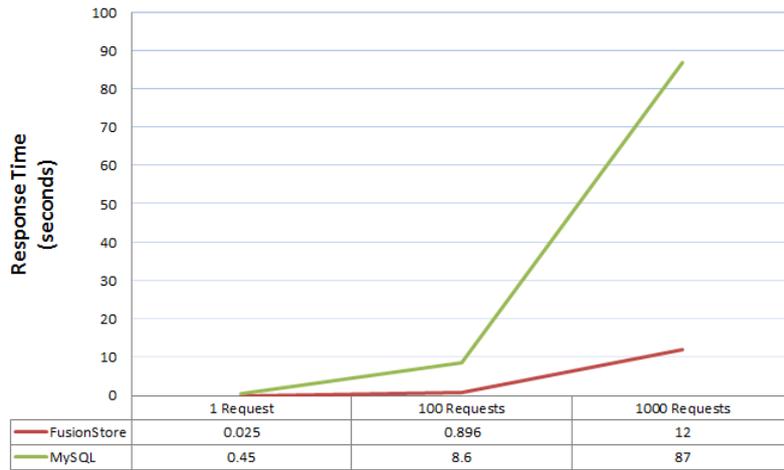
**Data Load:** 58 Seconds

**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:

### Query Response Time for 22k Series Fusion Store vs MySQL



## 9 Resync structures with Hub

The Node will periodically poll the hub service to ensure it is up to date, if any failures occur the Node should manage these by automatically re-syncing with the hub. However, if for any reason the structures in the Node are not in sync with the hub, they can be resynced by clicking the 'Re-Sync Structures with Hub' button in the web interface.

Alternatively, the structure file can be deleted to force a re-sync. The structure file can be found in the root folder of the Node file system.

*<Tomcat Home>/webapps/[FusionNode]/WEB-INF/node*