# DATA STORE MANAGEMENT

# Contents

# Version History

| Version # | Implemented By | Revision Date | Reason |
|---|---|---|---|
| 20190429 | Phil Lazarou | 24 April 2019 | Expanded the external database section |
| 20190510 | Phil Lazarou | 10 May 2019 | Further updates to external database section |
| 20190723 | Glenn Tice | 23 July 2019 | Clarify that the ID field for external database must be the SDMX Series Key |
| 20190813 | Kyle Nicholls | 13 August 2019 | Added sub agencies to externally managed data section |
| 20190821 | Kyle Nicholls | 21 August 2019 | Revised Section 2.4 Externally Managed Relational Databases |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# 1   Overview

Fusion Registry provides the facility of allowing Data Providers to report data.  Data may be reported in a number of ways: by Registering the URL of a data File; by Registering the URL of a valid SDMX REST web service; by publishing a Dataset to a data store; by re-indexing a database which has data loaded by a process run outside of the Fusion Registry.  The Fusion Registry can support any number of data collection methods, and each Dataflow may collect data in a number of ways, depending on the Data Provider.  In this way, the Fusion Registry is able to operate as a Data Portal over any number of disparate data stores.
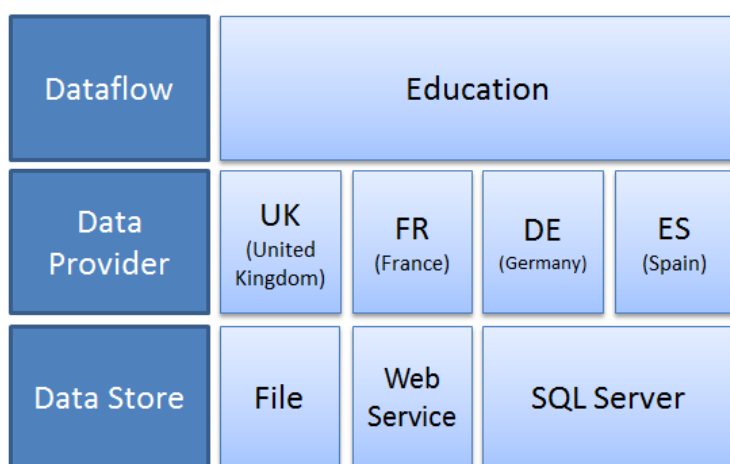


*Figure 1 showing an example of how a Dataflow with multiple Data Providers can have data split across multiple Data Stores*

The remainder of this document describes each data store type, how to set up connections to data stores, how to link a Data Provider to a data store, and how to manage reported data.

**Note:** The Data Reporting guide should be read for information on how to publish data and register the location of external files or web services.

## 2  Data Store Types

### 2.1  Overview

Data Stores can be split into two main types, Registry Managed Data Stores, and External Data Stores. A Registry Managed Data Store is a store for which the Fusion Registry is responsible for creating database tables, importing, managing and deleting content. Registry Managed stores includes the Fusion Data Store, and any MySQL or SQL Server database that has been configured to act as a Data Store. External stores include MySQL, SQL Server, or Oracle database which has its data managed by an external process, and SDMX web services or external files.

| Datasource | Registry Managed | | External | | | | |
|---|---|---|---|---|---|---|---|
| Data Store Type | Internal | Relational Database | Relational Database | | | URL | |
| Data Store | Fusion Data Store | My SQL | SQL Server | My SQL | SQL Server | Oracle | File | Web Service |

*Figure 2 showing the data stores supported by the Fusion Registry*

The Fusion Registry creates and maintains the link between a data store and the SDMX Structural Metadata at the level of a Provision Agreement. The Provision Agreement defines which data store it is connected to, defaulting to any external URL (File or Web Service). Each data store can have zero to many Provision Agreements linked to it. As both Dataflows and Data Providers can be referenced by multiple Provision Agreements, it is therefore possible to split data for the same Dataflow or same Data Provider over multiple data store types.

The notion that data exists is maintained by the Data Registration. The Data Registration references the Provision Agreement and defines the URL of where the data can be retrieved for the Provision Agreement. For data source type which maps to an external File or Web Service URL, it is the responsibility of the Data Provider to supply the URL of the service or file. For all other data source types, the Fusion Registry is responsible for generating, updating, and deleting the Data Registration, where the URL of these Data Registrations is to the Fusion Registry's own public web service.

The UML diagram below shows the relationship and cardinality between the SDMX Structures and the Fusion Registry data store types.
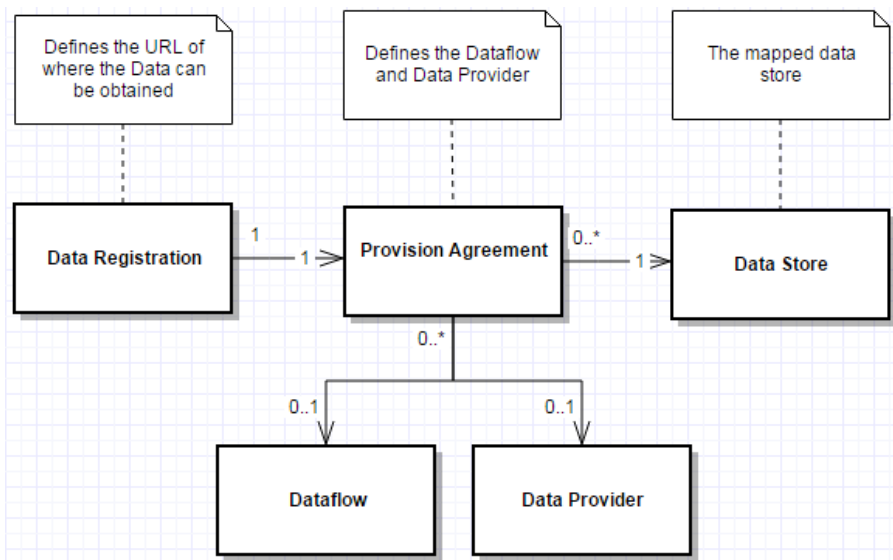
The remainder of this section describes each data store type in more detail.
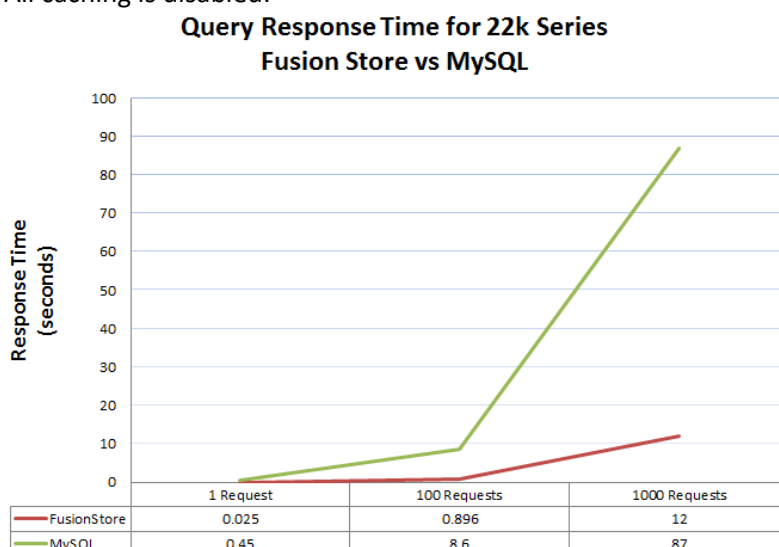
## 2.2 Fusion Data Store

The Fusion Data Store provides a high-speed in-memory database pre-installed in the Fusion Registry. Data published to this store is persisted to the database that the Fusion Registry is connected to, however all data is served from memory using a highly efficient in memory model.

Metrics for the Fusion Data Store are shown below:

**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 |

It should be noted that as the data queries are processed in memory, it is possible to horizontally scale the data tier by simply launching an additional Fusion Registry instance. Horizontally scaling the data tier ensures there is not a single point of failure, and increases the supported concurrency and therefore overall performance of the solution.

## 2.3 Fusion Registry Managed Relational Database

The Fusion Registry can be connected to any number of relational databases (MySQL or SQL Server) for the purpose of supporting data publishing. The Fusion Registry is responsible for building the database tables, which are built on demand when new data are published. Each Dataflow has its own database tables associated with it, and if they already exist when new data are published then the table is updated if required (if the DSD Concept, or Codelists change in a way which requires a modification).

The database tables in a Fusion Registry managed database are built based on the Data Structure Definition, and associated structures. Having database tables tailored to each data domain ensures the data load and retrieval is as efficient as possible.

The Fusion Registry managed database is not as fast as the Fusion Data Store and the performance is largely dependent on the database platform used (MySQL or SQL Server) and the installed hardware and network speed.

Unlike the Fusion Data store, the Fusion Registry managed database records modifications made to Observations values. These modifications are termed 'revisions' or 'vintages' and are made whenever an observation value is changed (revised) over time. As revisions are stored it is made available via the SDMX web services, using the 'includeHistory' parameter. This parameter is documented in the Registry Web Services Guide.
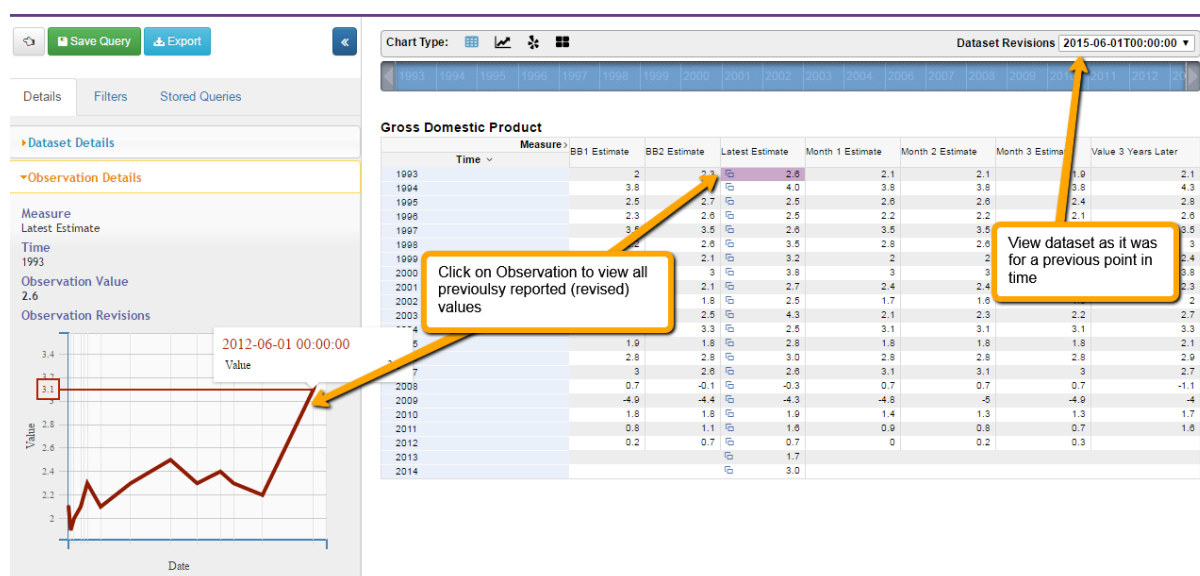


*Figure 3 showing an example of how revised Observation values are displayed in the Data Browser. The dataset in this example is Gross Domestic Product data from the UK's Office for National Statistics*

## 2.4   Externally Managed Relational Database

### 2.4.1   Overview

The Fusion Registry can connect to an Oracle, SQL Server or MySQL database in read-only mode.  This can be used when data already exists in a database which is managed by a process or application other than the Fusion Registry.

The Fusion Registry is able to query the database tables or views, provided they conform to the expected structure.  The table or view name and the column names are derived from the Dataflow that is linking to the data store.

There are two possible modes for mapping external database tables: "Default" and "Registry"

### 2.4.2   "Default" External mode

In this mode a single table or view should be created, the name of which must use the following pattern:

 *[Dataflow Agency Id]_[Dataflow Id]_[Dataflow version]*

The table name must be in upper case, and any '.' or '-' characters must be replaced with the underscore character '_'.  For example:

| Dataflow Agency | Dataflow Id | Dataflow Version | Expected Table/View Name |
|---|---|---|---|
| **WB** | POVERTY | 1.0 | WB_POVERTY_1_0 |
| **ONS.DEPT1** | GDP-DATA | 2.1.1 | ONS_DEPT1_GDP_DATA_2_1_1 |

**Table Structure**

The database table or view must contain a column for each of the Components of the Data Structure Definition that the Dataflow conforms to.  For time series data, an additional column with the name **FR_OBS_DATE** must be providing containing the date and time of the observation. For MySQL the column type should be "*datetime*", for Oracle: "*timestamp*".

Example:

| DSD Component ID | Component Type | Column Id | Column Type |
|---|---|---|---|
| **FREQ** | Dimension | FREQ | N/A |
| **REF_AREA** | Dimension | REF_AREA | N/A |
| **SERIES** | Dimension | SERIES | N/A |
| **SERIES_TITLE** | Series Attribute | SERIES_TITLE | N/A |
| **OBS_VALUE** | Observation Attribute | OBS_VALUE | N/A |
| **OBS_CONF** | Observation Attribute | OBS_CONF | N/A |
| **OBS_VALUE** | Observation Value | OBS_VALUE | N/A |
| **TIME_PERIOD** | Time Dimension | TIME_PERIOD | String (ISO-8601 Date) |
| **-** | - | FR_OBS_DATE | DateTime |

The FR_OBS_DATE column is used when performing data queries which have start and end date restrictions. However, the value in the TIME_PERIOD column must represent the same date, formatted according to the frequency of the series.  The following table shows an example of the observation frequency vs the expected date format.

| Frequency | Format | Example |
|---|---|---|
| Annual | YYYY | 2010 |
| Daily | YYYY-MM-DD | 2010-01-01 |
| Date Time | YYYY-MM-DD-Thh:mm:ss | 2010-01T20:22:00 |
| Monthly | YYYY-MM | 2010-01 |
| Quarterly | YYYY-Qn | 2010-Q1 |
| Semester | YYYY-Sn | 2010-S1 |
| Trimester | YYYY-Tn | 2010-T1 |

### 2.4.3 "Registry" External mode

In "Registry" External mode, the data must be presented in a normalised schema consisting of two tables or views in a one-to-many relationship.
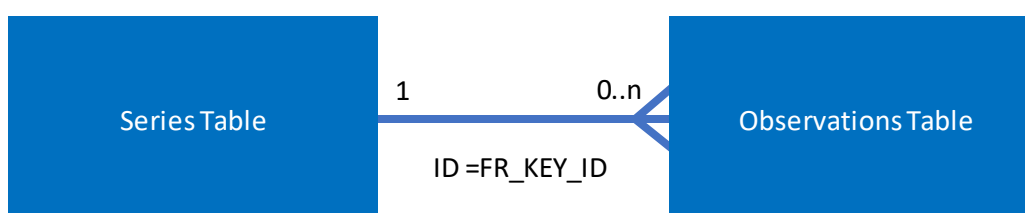


Each series appears only once in the Series table, uniquely identified by the 'ID' primary key. The observations are stored in the Observations table with 'FR_KEY_ID' acting as the series foreign key.

**Series Table**

The series table must conform to the following naming structure:

*[Dataflow Agency Id]_[Dataflow Id]_[Dataflow version]_key*

The table name must be in upper case, and any '.' or '-' characters must be replaced with the underscore character '_'. For example:

| DataFlow Agency | Dataflow Id | Dataflow Version | Expected Table/View Name |
|---|---|---|---|
| WB | POVERTY | 1.0 | WB_POVERTY_1_0_key |
| WB.STAT | GDP-DATA | 2.1.1 | WB_STAT_GDP_DATA_2_1_1_key |

Due to limitations imposed by some databases, the table name cannot exceed 30 characters. In this case, different rules are applied as described in Section 2.4.7.

The database table or view must contain:

- a column for each of the Dimensions of the Data Structure Definition that the Dataflow conforms to;
- an 'ID' column containing the SDMX Series Key - the Series Key is a colon (:) separated list of the series' dimension values.

The table or view can also contain an optional FR_UPDATED_DATE column which must be of a type that can support both Date and Time. For MySQL the columns need to be of type "*datetime*", for Oracle: "*timestamp*". If used, FR_UPDATED_DATE should either contain the date / time when the

series was last updated, or 'null' to indicate there are no observations. There's more about managing series with no observations in Section 2.4.4.

The following example shows the table Series Table columns required for a DSD with 3 dimensions:

| DSD Component ID | Table Column Name | Example Value |
|---|---|---|
| n/a | ID | A:AG_AGR_TRAC_NO:ABW |
| FREQ | FREQ | A |
| SERIES | SERIES | AG_AGR_TRAC |
| REF_AREA | REF_AREA | ABW |
| n/a | FR_UPDATED_DATE | 21-JUN-19 00.00.00 |

**Observations Table**

The Observations table must conform to the following naming structure:

*[Dataflow Agency Id]_[Dataflow Id]_[Dataflow version]_obs*

The table name must be in upper case, and any '.' or '-' characters must be replaced with the underscore character '_'.  For example:

| Dataflow Agency | Dataflow Id | Dataflow Version | Expected Table/View Name |
|---|---|---|---|
| WB | POVERTY | 1.0 | WB_POVERTY_1_0_obs |
| WB.STAT | GDP-DATA | 2.1.1 | WB_STAT_GDP_DATA_2_1_1_obs |

Due to limitations imposed by some databases, the table name cannot exceed 30 characters. In this case, different rules are applied as described in Section 2.4.7.

Each row of the Observations table holds information for a single observation with FR_KEY_ID containing the SDMX Series Key of the series.

The table must have the following structure:

| Column ID | Data Type | Purpose |
|---|---|---|
| FR_KEY_ID | String | Links to the "Series" table. The value must be the same as in the ID field of the Series table. |
| FR_OBS_DATE | DateTime | The date of the observation (in full date time form) |
| TIME_PERIOD | String | The date of the observation in String (ISO-8601 Date) |
| OBS_VALUE | String | The value of the observation |
| <column for each of the observation level attributes> | String | For each observation attribute, a column must be provided |
| FR_VALID_FROM | DateTime | Optional - the date / time from which the observation is considered to be valid |
| FR_VALID_TO | DateTime | Optional - the date / time to which the observation is valid to<br>The last valid observation should have the date 31-Dec-2999 00:00:00 |
| FR_UPDATED_DATE | DateTime | Optional – the date / time when the observation was last updated |

The FR_OBS_DATE, FR_VALID_FROM, FR_VALID_TO and FR_UPDATED_DATE columns must be of a type that can support both Date and Time. For MySQL the columns need to be of type "*datetime*", for Oracle: "timestamp".

FR_UPDATED_DATE is optional. If it exists, it should be populated with the date and time when the observation was last updated. Fusion Registry will use this information when responding to data queries with the 'updatedAfter' REST data query parameter as explained in Section 2.4.6.

### 2.4.4    DSDs with a Single Dimension

In "Registry" External mode, Series tables for DSDs with a single dimension need only the ID column which avoids having two columns (ID and the dimension column) with exactly the same values.

A common use case for DSDs with a single dimension is modelling time series where each series is uniquely identified by a series code or indicator.

### 2.4.5    Series with no Observations

In "Registry" External mode, series can be listed in the Series table without having observations in the Observations table. For such series, the FR_UPDATED_DATE column should be set to 'null'.

When observations are added, FR_UPDATED_DATE must be set to the appropriate date/time otherwise the observations will be ignored.

Series without observations where FR_UPDATED_DATE is not 'null' will be indexed by Fusion Registry as *empty series*. Empty series are discoverable by data consumers and will be returned in query results. This may be desirable in certain circumstances, but consideration should be given as to the intended behaviour to avoid end-users unexpectedly receiving empty series when querying for data.

### 2.4.6    Behaviour of Queries Using the 'updatedAfter' REST API Request Parameter

REST API data queries can include the 'updatedAfter' request parameter to select only data that was updated after the specified time. Fusion Registry uses the following procedure when responding to such queries where the dataset is held in an externally managed relational database:

1.  If the Observation table has a FR_UPDATED_DATE column, the query result will contain only observations where the observation FR_UPDATED_DATE > [updatedAfter request parameter].

2.  Else, if the Series table has a FR_UPDATED_DATE column, the query will result will contain all observations for series where the series FR_UPDATED_DATE > [updatedAfter request parameter].

3.  Else, if neither the Observation or Series tables have an FR_UPDATED_DATE column, the query result will contain only observations where the observation FR_VALID_FROM > [updatedAfter request parameter].

4.  Else, if the Observation table does not have a FR_VALID_FROM column, the 'updatedAfter' parameter is ignored, and the result set will contain all series / observations matching the query.

### 2.4.7    Long Table Names

Some SQL databases impose restrictions on the length of table or view names. If the names of Series or Observation tables would exceed 30 characters, then Fusion Registry uses an algorithm to generate unique table names.

The table names for any provision agreement can be discovered through the Data Sources Manager using the following procedure:
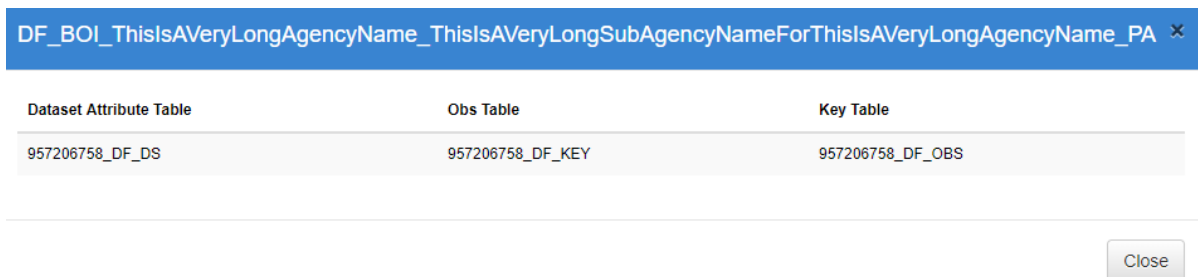
1. As an Administrator, go to the Data Sources Manager page.

2. Click the 'cogs' icon and select 'Link Provision Agreement'
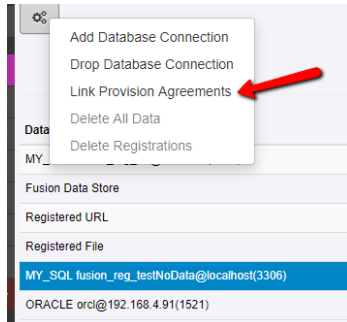


3. On the Provision Agreements window displayed, choose the required provision agreement and use the 'Table Names' button.



4. The names of the tables expected for the provision agreement are displayed:



Note that the Dataset Attribute table shown is used where the DSD includes dataset level attributes. Please contact Metadata Technology Technical Support for advice and guidance on this use case.

## 2.5   File Data Source

It is possible to define the data source for a Provision Agreement to be of type File.  If a data source is set to 'file', then it is up to the Data Provider to Register the URL of the file location.  This can be achieved by using the HTML User Interface to register a new data file, or by submitting a SDMX data Registration to the SDMX web service.

The file can be in any of the formats that are supported by the Fusion Registry.  For more information on supported File Formats, please read the Registry Data Formats guide.

## 2.6   Web Service Data Source

It is possible to define the data source for a Provision Agreement to be of type Web Service.  A Web Service data source expects the Data Provider to Register the URL of the web service entry point.  This can be achieved by using the HTML User Interface to register a new data file, or by submitting a SDMX data Registration to the SDMX web service.  The registered web service must conform to the SDMX REST specification, as documented in Section 7 of the SDMX specification.  The Fusion Registry will

index the web service on registration by performing an upfront query to the web service for all series keys for the given Dataflow. The upfront query will be a query for all data, with detail set to seriesKeysOnly. For example:
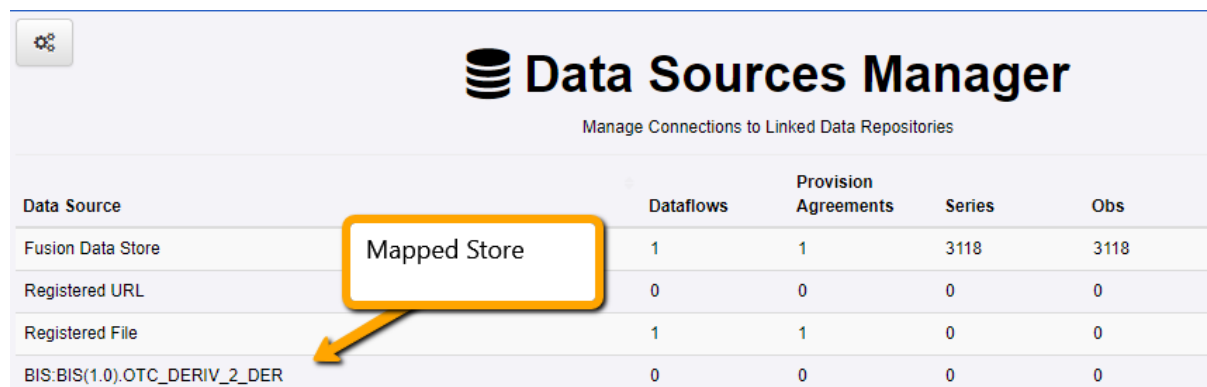
http://www.demoservice.org/ws/**data/FLOW_ACY,FLOW_ID,FLOW_VERSION/all/DP_ACY,DP1?detail=seriesKeysOnly**

The registered web service must be able to respond to this data query in order for the registration to succeed.

## 2.7   Mapped Data Store

A mapped Data Store is used when the data is stored conforming to one Data Structure Definition, but disseminated conforming to another Data Structure Definition. To make use of a Mapped Data Store, there must first be a definition of the mapping, the Structure Mapping documentation provides full details on structure maps, and should be read prior to this section.

Once a Structure Map is defined describing the relationship between a source and target Dataflow, it will appear in the list of available data source from the **Data Manager** page.



*Figure 4 showing a mapped store. The Store Name is the Agency Id, Id, and version of the Structure Map which defines the mapping*

A Structure Map defines a source and target Dataflow, for example:



If a Provision Agreement for the **POPULATION** Dataflow is linked to the Mapped Store, then the Fusion Registry will convert an incoming data query for POPULATION to conform to a data query for the **FAME_POP** Dataflow. The data query will be executed against the **FAME_POP** dataflow, and the data query response will be mapped on the way out to conform to the requested **POPULATION** data structure.

In this example, in order for the mapping to bring back data, the **FAME_POP** dataflow should be linked to a data store, and have data loaded. Each time a data registration occurs for the **FAME_POP** dataflow, the **POPULATION** dataflow will have its indexes updated. A data registration can occur as a

result of loading data (if the store is Registry managed) or a manual data registration (if the store is not registry managed, i.e File or URL or external database).

When a provision agreement is using a mapped data store, the data itself will always reside in the target dataflow's store.  The data will be mapped on-the-fly.  An example workflow is shown in the image below.
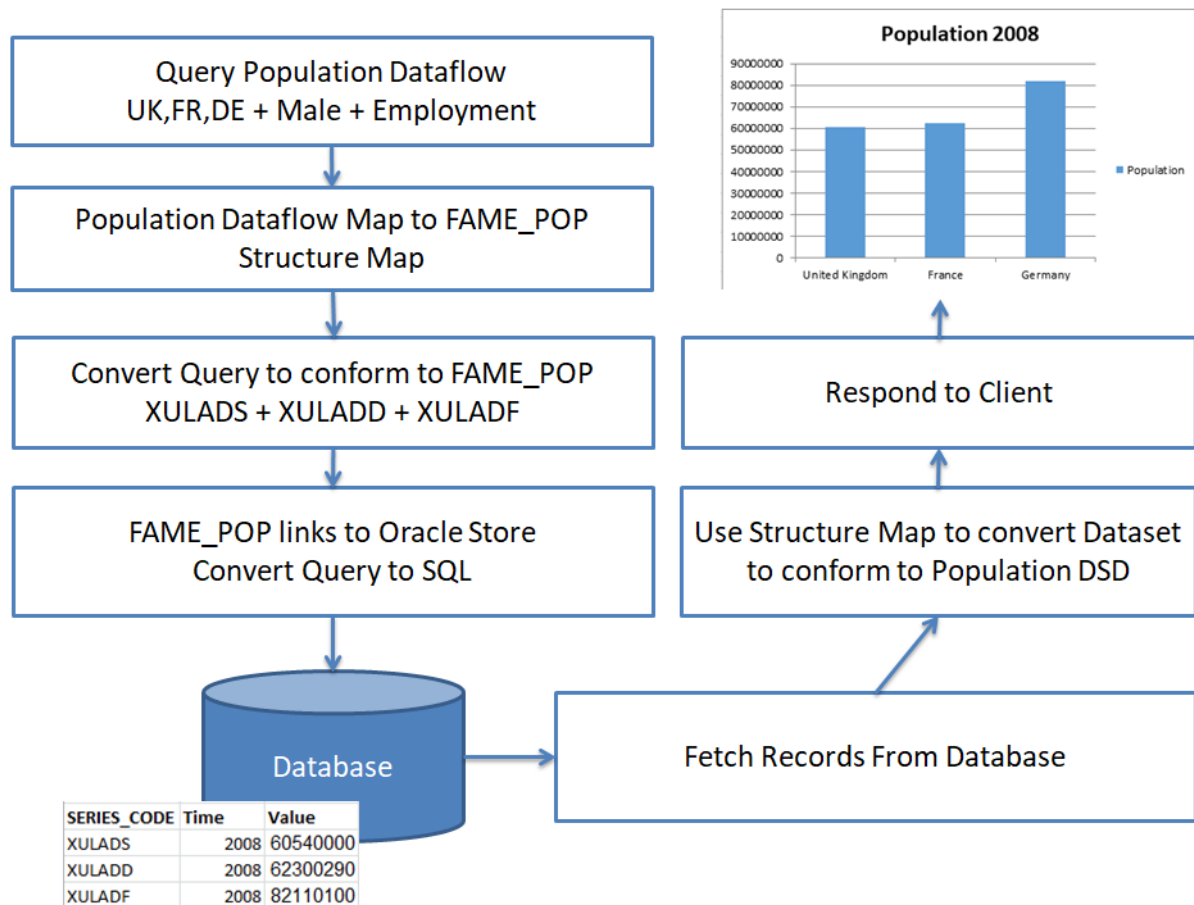


*Figure 5 showing the mapping workflow from data query to data response where the Population Dataflow is linked to a mapped store, the target dataflow for the mapped store (defined by the associated structure map) is FAME_POP which itself links to a relational database.  The client remains agnostic to the mapping, and can retrieve the data in any Fusion Registry supported data format.*

# 3 Data Sources Manager

## 3.1 Overview
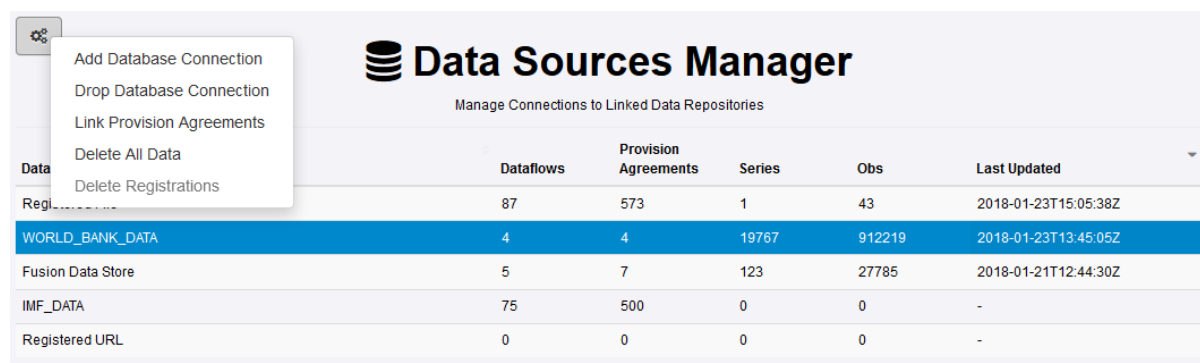
The Data Sources Manager is accessed by an Admin user in the Fusion Registry Web Interface. The Data Sources Manager provides information on the data located in File, Web Service, and the Fusion Data store. In addition it also provides the means to create a connection to both Registry Managed and External databases.

| Data Source | Dataflows | Provision Agreements | Series | Obs | Last Updated |
|---|---|---|---|---|---|
| Registered File | 87 | 573 | 1 | 43 | 2018-01-23T15:05:38Z |
| WORLD_BANK_DATA | 4 | 4 | 19767 | 912219 | 2018-01-23T13:45:05Z |
| Fusion Data Store | 5 | 7 | 123 | 27785 | 2018-01-21T12:44:30Z |
| IMF_DATA | 75 | 500 | 0 | 0 | - |
| Registered URL | 0 | 0 | 0 | 0 | - |

*Figure 6 showing data stores, including built in data stores and linked databases (IMF_DATA, WORLD_BANK_DATA)*

The Data Sources Manager provides an overview of all the available data sources in the Fusion Registry along with a count of how many Provision Agreements are linked to the data source and a total count of available Series and Observations over all the Provision Agreements.

The Data Sources Manager provides the means to bulk add or remove the connection of Provision Agreements to Data Stores. Individual Series or Observations can be deleted from Registry Managed data stores, external URLs can be deleted in bulk.



*Figure 7 showing the actions available in the Data Sources Manager page*

The remainder of this section describes the functionality of each of the actions in the image above.

## 3.2 Adding a New Database Connection

A database connection establishes a link to a Relational Database (MySQL, Oracle, SQL Server) for the purpose of exposing the data in a read only externally managed database, or creating a Registry Managed connection.

To create a new Database Connection, first click the Maintenance Button in the Data Sources Manager page as shown below.

Clicking on the Add button will open a form which must be completed to establish the database connection, this is shown in the image below.

*Figure 8 showing the Add Data Source modal*

The Connection Id is used entirely for identification purposes within the Registry itself, it must however contain no spaces and only consist of alphanumeric characters and underscores. The Connection Type refers to if the database connection is for the purpose of a Registry Managed database or an Externally Managed database. If it is Registry Managed then database tables are built on demand as data are loaded. If it is Externally Managed then database tables are expected to exist for any Provision Agreement which is linked to the Data Store.

The remainder of the form is used to establish the connection to the database. Single Sign On (SSO) is supported for SQL Server databases. SSO uses the authentication credentials of the user who launched the Tomcat server to authenticate with the SQL Server database. The Setup Guide provides more information about how to configure the Fusion Registry for SSO support.

## 3.3   Linking Provision Agreements

A Provision Agreement is linked to a Data Store to inform the Fusion Registry where the data for that Provision Agreement should be obtained from. If the Provision Agreement is linked to a Registry Managed data store then the action of linking allows data to be published to the data store. If there is already in the data store on link, then the data will be indexed and made available for query.

**Note:** If the Provision Agreement is linked to a read only relational database, then it will be indexed on link, and the data will be made available for query. Re-indexing a read only store is required when data is updated in the data store. Re-Indexing can be performed from the Data Dashboard page, by selecting the provision agreement and clicking the 'Re-Register' button. An automated process can be set up, by submitting a Data Registration to the web service. To generate a Data Registration SDMX file, click the 'Automate Registration' button in the Data Dashboard.

Linking to a Web Service or File data store allows Data Providers to register the URL of the data web service or file. All data publication and registration details are discussed in the Data Providers Guide.

Provision Agreements can be linked to a data store via the Dataflow Wizard, or via the Provision Agreement page, this is discussed in the Structural Metadata Manager Guide.  The Data Source

Manager page allows multiple Provision Agreements to be linked to a data store in one action. A Provision Agreement can only link to one data store, so the action of linking a Provision Agreement to one data store will unlink it from another. The action of unlinking from a relational database (both registry managed and read only) will not delete the data from the database. The action of unlinking from web service or file stores will delete any Data Registrations made against the Provision Agreements and therefore relinking will require a re-registration. Unlinking from the Fusion Data Store will delete the data from the Fusion Data Store.

## 3.4   Data Rollback

If using the Fusion Data Store, data submissions can be rolled back to a previous state. The last 5 submissions are stored for each Provision Agreement.

To rollback data to a previous version, the Data Sources Manager page allows data submitted against a Fusion Data Store to be rolled back.



*Figure 9 showing the Data Sources Manager*

Within the Data Sources Manager, selected the appropriate Provision Agreement and a table of updates will be displayed in the lower right of the screen. This will show the submission time, action and number of series in each data submission.

Rolling back the data submission will create a new submission of action "Replace" which will restore the data to the selected point in time. This makes it possible to roll back a roll-back.

**Note:** data rollbacks are only possible against data stored in a Fusion Data Store.

## 3.5   Drop Connection

Dropping a connection to a database will not delete any data in the database, however the stored connection details will be deleted and the active connection will be dropped. Data from the data store will no longer be accessible from the Fusion Registry user interface or web service.

Any Provision Agreement linked to the data store will be unlinked and linked to the data store type "Web Service or File". The series count for each unlinked Provision Agreement will revert to zero.

Any Data Registrations for Provision Agreements linked to the data store, will be deleted.

## 3.6   Delete all Data

The Delete all data action is only possible for a Registry Managed data store.  The effect of deleting all data in a data store will drop all database tables related to any linked Provision Agreements.  The Provision Agreements will still be linked to the data store, however the series count will revert to zero.

Any Data Registrations for Provision Agreements linked to the data store, will be deleted.

## 3.7   Delete Registrations

The Delete Registrations action is only possible for a Web Service or File data store.  On selecting this action a table will be displayed showing all Data Registrations against the store, with the option to select any number of Data Registrations to delete.  The action of deleting a Data Registration will mean the series count indexes will be updated and the data which was accessible from the deleted Registrations, will no longer be available via the web service or user interface.

## 3.8   Edit Data

The Edit Data button opens a data query window as shown in the image below.  On selecting an Observation cell in the data table that results from applying the query filter, the actions to delete series or observations will become available.



*Figure 10 showing the delete Observation and Series functionality*

# 4 Monitoring Data Publication Activity

## 4.1 Viewing Historical Data Registrations

It is possible to view historical Data Registrations in the Fusion Registry User Interface (UI). Authentication is not necessary to access the basic details via the UI, however if logged in as an Admin user extra details are available such as the username of the authenticated user who actioned the Data Registration, and the ability to view full request and response details including the associated server logs for the request.

It is important to note that a Data Registration event occurs each time data changes in the Fusion Registry, this may be due to a Data Provider registering the URL of an external data source, it may be due to data being imported or deleted from a Registry managed database, or the re-index of a read only data store.

To view Data Registration Activity click on the Activity menu item on the left hand menu, followed by Data Registrations, as shown below.



*Figure 11 showing the Data Registration Activity page*

Data Registration information can be broken down in different ways, the upper table contains the high level breakdown, and the lower table shows the Data Registrations, this is shown in the above image.

## 4.2 Subscribing to Changes

It is possible to subscribe to Data Registration events in the Fusion Registry, this can be achieved by clicking the 'subscribe to changes' button in the lower right of the user interface (in the footer). Subscriptions can be added for all Data Registration Events, or for specific Dataflows/Data Providers. An email address is required for a Subscription this email address is required to load the Subscription for future maintenance, the Subscription Notification message can be sent to the same email address and/or can be POSTed via HTTP to a given URL.

18

Data Registration notifications include the SDMX Data Registration XML in the email, as well as the POST message.

## 4.3   RSS Feed

The Fusion Registry provides a RSS feed which is updated every time a structure or registration event occurs in the Fusion Registry.  This includes creation, updates, and deletions.  The RSS feed is linked to from the Fusion Registry home page by clicking on the RSS icon: .