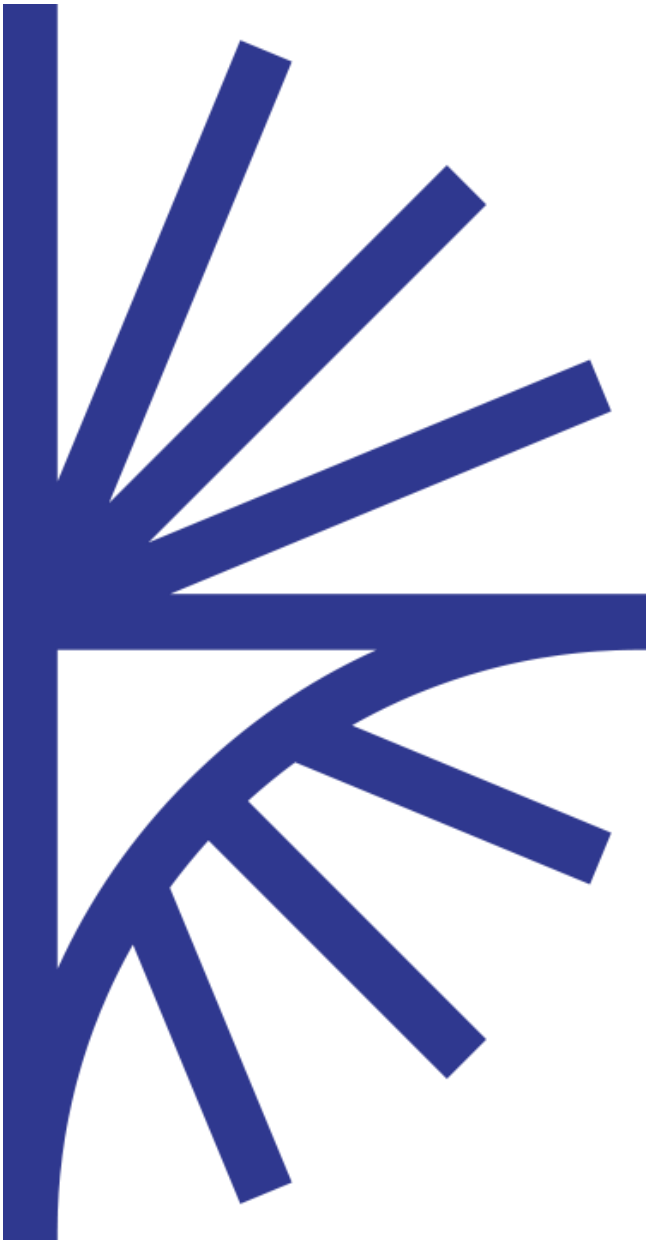


FUSION REGISTRY

Data Formats

FUSION REGISTRY
VERSION 9

Data Formats

This guide documents the supported Data Formats for import and export in the Fusion Registry

Fusion Registry: 9.5.x
Document Last Updated: April 27th 2018

Version 20180427

1 Contents

2 Overview

This document provides an overview of all data formats supported as import by the Fusion Registry, additional data formats are supported for export (such as RDF formats).

For SDMX data formats, additional documentation is available from sdmx.org.

The Fusion Registry includes a data samples directory in the distribution, which contains the same dataset described for each of the formats contained in this document.

It is important to note that this document discusses the advantages of some of the formats with regards to the level of validation that can be achieved by an XML Schema (XSD), the Fusion Registry however uses its own data validation processes to give a deeper level of validation then an XML schema can perform. The Fusion Registry data validation processes are the same regardless of data import format, and therefore the notes in this document about schema validation are for information only. The chosen data import format has no impact on the level of validation that will be performed on the imported dataset.

It is also important to note that this document describes the supported import format that come as standard in the Fusion Registry distribution. The Fusion Registry makes it possible to implement support for new data formats for both import and export.

3 SDMX Data Formats

3.1 Generic Data

Generic Data is in all versions of the SDMX Standard, each version is supported by the Fusion Registry including GenericTimeSeriesData which was introduced in version 2.1 of SDMX.

A Generic Dataset is defined by the SDMX Schemas (xsd files) distributed in section 3B of the standard. A Generic Dataset contains the same XML nodes and attributes regardless of the data it is transmitting, hence the name 'generic'. Below is an example of a series in a Generic Dataset.

The SDMX Schema can validate a Generic Dataset but it cannot be used to verify any values in the reported dataset.

```
<generic:Series>
  <generic:SeriesKey>
    <generic:Value id="GEO" value="AT"/>
    <generic:Value id="SEX" value="T"/>
    <generic:Value id="HST" value="NPRV"/>
    <generic:Value id="LMS" value="DIV"/>
    <generic:Value id="CAS" value="POP"/>
    <generic:Value id="POB" value="TOTAL"/>
    <generic:Value id="COC" value="TOTAL"/>
    <generic:Value id="AGE" value="TOTAL"/>
    <generic:Value id="FREQ" value="A"/>
  </generic:SeriesKey>
  <generic:Attributes>
    <generic:Value id="TIME_FORMAT" value="P1Y"/>
  </generic:Attributes>
  <generic:Obs>
    <generic:ObsDimension value="2011"/>
    <generic:ObsValue value="10488.0"/>
    <generic:Attributes>
      <generic:Value id="OBS_STATUS" value="I"/>
      <generic:Value id="OBS_NOTE" value="LMS: Including persons whose same-sex registered partnership was legally dissolved."/>
    </generic:Attributes>
  </generic:Obs>
</generic:Series>
```

Figure 1 showing a single series for a Generic Dataset

3.2 Structure Specific / Compact Data

Structure Specific Data is a SDMX v2.1 re-branding of the term Compact Data which existed in versions 1.0 and 2.0 of the SDMX specification. The Fusion Registry supports each version including StructureSpecificTimeSeriesData which was introduced in version 2.1 of the SDMX specification.

Section 3B of the SDMX distribution includes the abstract schema (XSD) definition for a Structure Specific dataset, but the concrete implementation is dependent on what Provision Agreement, Dataflow, or Data Structure Definition the dataset is for.

The Fusion Registry is able to generate a Structure Specific XSD on request from the web services page.

As the generated Schema is specific to the structural metadata held in the Fusion Registry, the level of validation offered by the schema is far greater then with a Generic dataset. A Structure Specific dataset is also more terse then a Generic dataset, as shown in the example below.

```
<Series GEO="AT" SEX="T" HST="NPRV" LMS="DIV" CAS="POP" POB="TOTAL" COC="TOTAL" AGE="TOTAL" FREQ="A" TIME_FORMAT="P1Y">
  <Obs TIME_PERIOD="2011" OBS_VALUE="10488.0" OBS_STATUS="I" OBS_NOTE="LMS: Including persons whose same-sex registered partnership wa
</Series>
<Series GEO="AT" SEX="T" HST="NPRV" LMS="MAR" CAS="POP" POB="TOTAL" COC="TOTAL" AGE="TOTAL" FREQ="A" TIME_FORMAT="P1Y">
  <Obs TIME_PERIOD="2011" OBS_VALUE="17811.0" OBS_STATUS="I" OBS_NOTE="LMS: Including persons in a same-sex registered partnership."/>
</Series>
```

Figure 2 showing an example of two series in Structure Specific / Compact format

3.3 SDMX-EDI

SDMX-EDI is a 1990s syntax that uses the UN/EDIFACT syntax. It is an extremely terse data format that supports time series only. The format is still popular for reporting financial data in the Central Banking community. However, unless you have a specific requirement to use this format then it is not recommended as the syntax is largely superseded by 21st century syntaxes such as XML and JSON.

For more information on the structure and syntax of SDMX-EDI, please visit sdmx.org.

```
ARR++AT:T:NPRV:DIV:POP:TOTAL:TOTAL:TOTAL:A:2011:602:10488.0:I'
ARR++AT:T:NPRV:MAR:POP:TOTAL:TOTAL:TOTAL:A:2011:602:17811.0:I'
```

Figure 3 showing an example of two series represented in SDMX-EDI format

3.4 SDMX-JSON

JSON stands for Java Script Object Notation and this is a representation of the dataset in Java Script that can be processed by JavaScript applications. SDMX-JSON was introduced in version 2.1 of the SDMX Specification. The textual representation of the dataset components (Concepts and Codes) is presented in addition to its coded form. Therefore there is no need to access independently the DSD in order to process the datasets for visualisation. SDMX-JSON was originally envisaged as an export format to support web dissemination; however it is a supported data import format to the Fusion Registry.

For more information on the structure and syntax of SDMX-EDI, please visit sdmx.org.

```
"dataSets": [
  {
    "action": "Information",
    "annotations": [],
  "observations": {
    "0:0:0:0:0:0:0:0:0:0": [
      "10488.0", 0, null, 0, null, 0
    ],
    "0:0:0:1:0:0:0:0:0:0": [
      "17811.0", 0, null, 1, null, 0
    ],
    "0:0:0:2:0:0:0:0:0:0": [
      "57252.0", null, null, null, null, 0
    ],
    "0:0:0:3:0:0:0:0:0:0": [
      "39761.0", 0, null, 2, null, 0
```

Figure 4 showing an example of series in a SDMX-JSON dataset

4 Excel

4.1 Format

An Excel data file should contain 2 distinct sections: the header section and the data section.

The header section contains high level information about the dataset such as the Provision Agreement, Dataflow and Data Structure to which the data conforms, as well as any Dimensions or Attributes which contain fixed values for the entire dataset. The Header section may also contain additional information which will be ignored by the read process, for example in the image below, row 2 will be ignored.

	A	B	C
1	**Dataflow**	UIS:EDUCATION_MDG(1.0)	
2	**Dataflow Name**	Education Millenium Development Goals	
3	**FREQ**	A	
4	**SERIES**	LR_AG15T99	

Figure 5 showing part of the header section of a Excel dataset

The data section contains the observation values for the dataset and may also contain dimensions or attributes which are not necessarily fixed values.

The format of data in excel is expected to follow the following convention:

Worksheet Link to Structure

The header section can optionally provide the information about the linked Provision Agreement, Dataflow, or Data Structure. This is achieved by adding a cell in the column 'A' with the associated value which is either the URN without the prefix (if cell A defines which structure it is), or the fully qualified URN. The allowed values for row A, with an example value for the associated cell in row B is shown in the table below.

Value	Example	Description
DataStructure	UIS:MDG(1.0)	A postfix URN (everything following the equals sign in a URN). The postfix URN format is: <agency>:<id>(<version>)
Dataflow	UIS:EDUCATION_MDG(1.0)	
ProvisionAgreement	UIS:EDUCATION_BLZ(1.0)	
StructureURN	urn:sdmx:org.sdmx.infomodel.datastructure.Dataflow=UIS:EDUCATION_MDG(1.0)	A fully qualified URN to either a Data Structure, Dataflow, or Provision Agreement

Dimension, Attributes and Time Periods as header columns

	A	B	C	D	E	F	G	H	I
1	INDICATOR	FREQ	BASE_PER	UNIT_MULT	TIME_FORMAT	2001	2002	2003	2004
2									

Figure 6 demonstrating rule #1 – Dimensions, Attributes, and Time Periods as Column Headers

Dimensions and Attributes can have fixed values

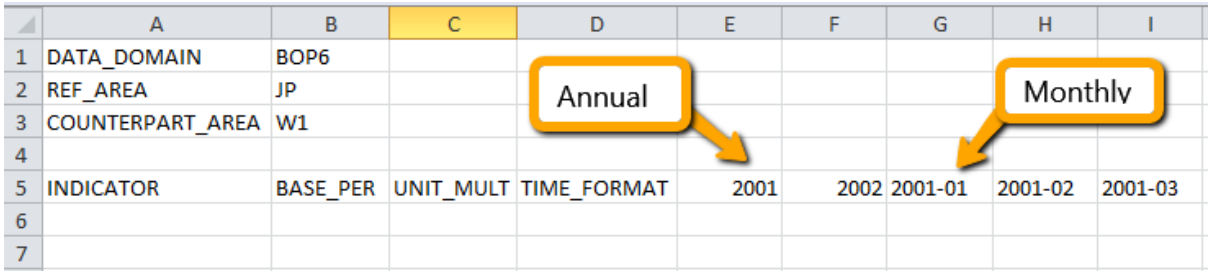
If any Dimensions or Attributes have fixed values for the whole dataset, these may be placed in a Header section. Shown below are fixed values for DATA_DOMAIN, REF_AREA and COUNTERPART_AREA:

	A	B	C	D	E	F	G	H	I
1	DATA_DOMAIN	BOP6							
2	REF_AREA	JP							
3	COUNTERPART_AREA	W1							
4									
5	INDICATOR	FREQ	BASE_PER	UNIT_MULT	TIME_FORMAT	2001	2002	2003	2004
6									

Figure 7 demonstrating rule #2 providing fixed Dimension and Attribute values for the dataset

Multiple Frequencies are supported

It is possible to report data against different frequencies[1] (Annual and Monthly for example). The frequency of the reported values is derived from the date formats.



	A	B	C	D	E	F	G	H	I
1	DATA_DOMAIN	BOP6							
2	REF_AREA	JP							
3	COUNTERPART_AREA	W1							
4									
5	INDICATOR	BASE_PER	UNIT_MULT	TIME_FORMAT	2001	2002	2001-01	2001-02	2001-03
6									
7									

Figure 8 showing multiple frequencies being reported

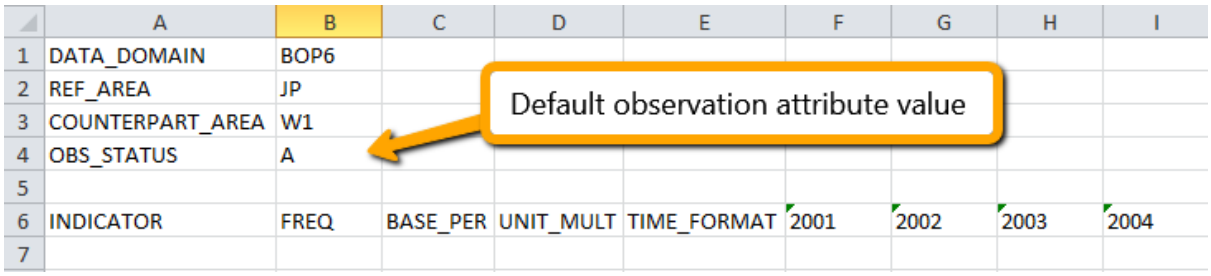
When the data is read, the value for the dimension FREQ will be derived from the corresponding date format. The following rules are used:

FREQ	Frequency	Format	Example
A	Annual	YYYY	2010
M	Monthly	YYYY-MM	2010-01
D	Daily	YYYY-MM-DD	2010-01-01
I	Date Time	YYYY-MM-DD-Thh:mm:ss	2010-01T20:22:00
Q	Quarterly	YYYY-Qn	2010-Q1
S	Semester	YYYY-Sn	2010-S1
T	Trimester	YYYY-Tn	2010-T1

Table 1 showing derived FREQ code Id against the format of the date String in the date column

Observation Attributes can have a Default Value

If applying default values to Observation Attributes, these may also appear in the header section.



	A	B	C	D	E	F	G	H	I
1	DATA_DOMAIN	BOP6							
2	REF_AREA	JP							
3	COUNTERPART_AREA	W1							
4	OBS_STATUS	A							
5									
6	INDICATOR	FREQ	BASE_PER	UNIT_MULT	TIME_FORMAT	2001	2002	2003	2004
7									

Figure 9 demonstrating rule #3 applying a default value for an observation attribute

[1] This can only be achieved if the Data Structure Definition has a FREQ Dimension, and the allowable values follow SDMX conventions.

Observation Values reported in Data Section

Reported values appear in the data section as shown below:

	A	B	C	D	E	F	G	H	I
1	DATA_DOMAIN	BOP6							
2	REF_AREA	JP							
3	COUNTERPART_AREA	W1							
4	OBS_STATUS	A							
5									
6	INDICATOR	FREQ	BASE_PER	UNIT_MULT	TIME_FORMAT	2001	2002	2003	2004
7	BCA_BP6_XDC	A		0	P1Y	12.2	22.2	32.2	42.2
8	BXCA_BP6_XDC	A		0	P1Y	12.3		32.3	42.3

Figure 10 demonstrating #4 reported values for Dimensions, Attributes and Time Periods

Extra Rows and Columns in the Spreadsheet

To improve the readability for the user it is often useful to have blank columns or rows in your spreadsheet. Blank rows are permitted but with certain restrictions. Blank rows may appear before the header section and between the header section and data section. However a blank row may not exist within the header section or the data section. If a blank row is encountered in the header section then this is assumed to indicate the end of the header section and this may cause your spreadsheet to be read incorrectly. In the data section a blank row indicates the end of the data section. If your data section contains blank rows then processing of the data will stop prematurely.

Blank columns also indicate that no further information should be read from that row. Fusion Registry will read from the first column of information in a row until it reaches a blank cell (unless it's a data row). The image below shows a spreadsheet where both column H and row 9 are blank. This would mean that in the data section only data for 2001 and 2002 is read (columns F and G) and because row 9 is blank, Fusion Registry will stop processing after row 8, therefore row 10 will not be processed.

	A	B	C	D	E	F	G	H	I	J
1	DATA_DOMAIN	BOP6								
2	REF_AREA	JP								
3	COUNTERPART_AREA	W1								
4	OBS_STATUS	A								
5										
6	INDICATOR	FREQ	BASE_PER	UNIT_MULT	TIME_FORMAT	2001	2002		2003	2004
7	BCA_BP6_XDC	A		0	P1Y	12.2	22.2		32.2	42.2
8	BXCA_BP6_XDC					12.3			32.3	42.3
9										
10	BEFD_BP6_XDC	A		0	P1Y	12.4	22.4		32.4	42.4

Figure 11 demonstrating the effect blank rows and columns have on the processing of data

It is permissible to have columns in the data section that are not dimensions, attributes or data but may contain additional information for the reader of the spreadsheet. The image below shows a spreadsheet where column F is for additional notes for each data row. The presence of this column will not prevent the data (in columns G to J) from being read even though the data rows themselves do not have a value for the row.

	A	B	C	D	E	F	G	H	I	J
1	DATA_DOMAIN	BOP6								
2	REF_AREA	JP								
3	COUNTERPART_AREA	W1								
4	OBS_STATUS	A								
5										
6	INDICATOR	FREQ	BASE_PER	UNIT_MULT	TIME_FORMAT	Additional Notes	2001	2002	2003	2004
7	BCA_BP6_XDC	A		0	P1Y		12.2	22.2	32.2	42.2
8	BXCA_BP6_XDC						12.3		32.3	42.3
9	BEFD_BP6_XDC	A		0	P1Y		12.4	22.4	32.4	42.4

Figure 12 demonstrating a column (column F) that is not data, dimension or attribute but will not prevent data processing

In order to assist in the generation of valid Excel data file, the Fusion Registry provides a mechanism to generate an Excel Template. The excel template is an Excel file with prefilled cells based on the selected Dataflow and Reporting Period. The second approach to authoring data makes use of a plugin library for Microsoft Excel to facilitate the construction of a dataset, by generating the column headers, and then providing an author helper which can be used to choose from a list of predefined values when filling in each column.

Support for Multiple Worksheets

Multiple worksheets are supported, however the subsequent worksheets must include a Header Section which links the data to the same structure as the first worksheet. Other fixed or variable values can be different, as shown in the two images below, where the second worksheet is reporting data for a different Frequency.

	A	B
1	**Dataflow**	UIS:EDUCATION_MDG(1.0)
2	**Dataflow Name**	Education Millenium Development Goals
3	**FREQ**	A

Figure 13 showing the header section for the first worksheet

	A	B
1	**Dataflow**	UIS:EDUCATION_MDG(1.0)
2	**Dataflow Name**	Education Millenium Development Goals
3	**FREQ**	Q

Figure 14 showing the header section for the second worksheet

4.2 Excel Template

To create an Excel Template, click on the Convert or Publish button from the home page, this takes the user to the Load Data page, from where the Excel Support button can be clicked.

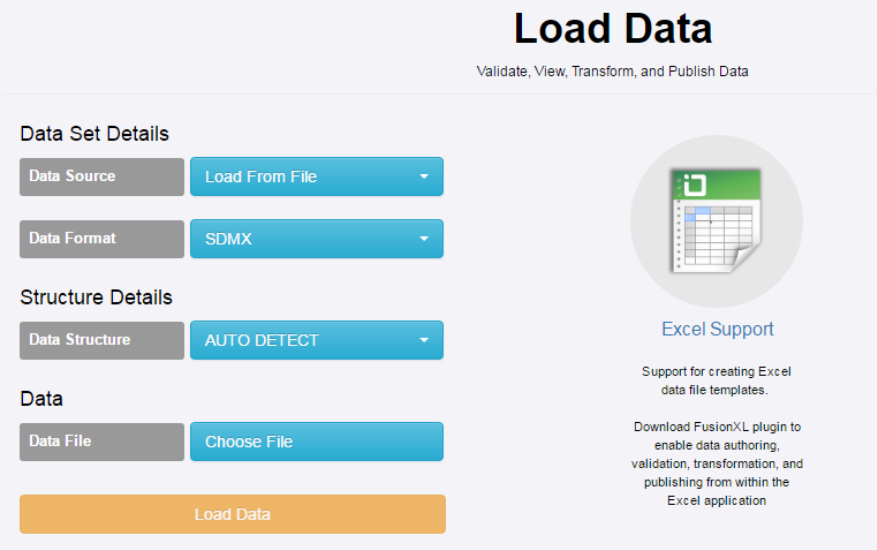


Figure 15 showing the Load Data Page with the Excel Support button

This will open up a new window, showing a list of all the available Dataflows, or if logged into the system, then the list will show the list of available Provision Agreements. A Provision Agreement is a Dataflow for which the Data Provider has been configured to provide data for. The recommendation is to log in first, as the Provision Agreement list will be shorter, and contain only the relevant Dataflows. Additionally the generated template may provide default values for the Data Provider.

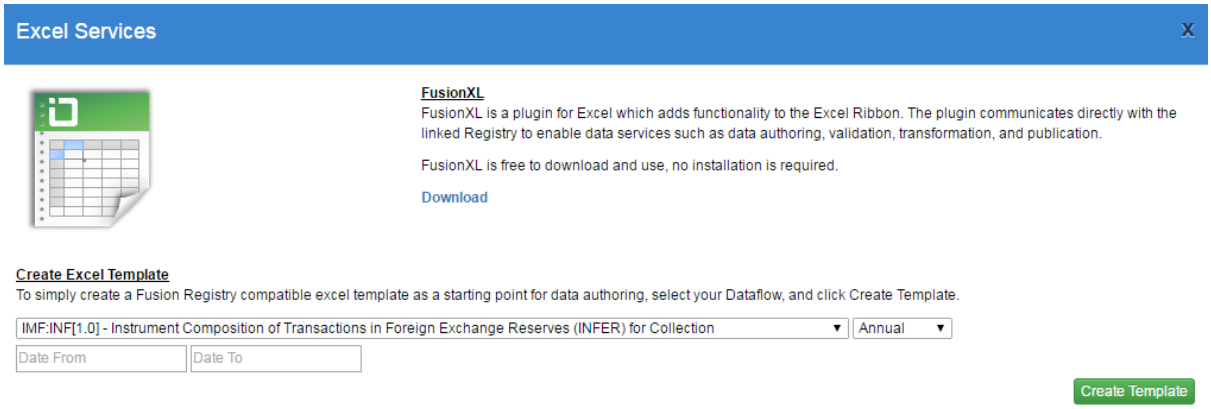


Figure 16 showing the Excel Services modal with the Create Excel Template form

To create an Excel Template, fill in the form, which includes the Dataflow/Provision Agreement, the frequency of data, and the data from and to dates. On clicking Create Template, and Excel file will be generated and downloaded to the user's local machine. The Excel template will have all columns prefilled based on the completed form information.

	A	B	C	D	E
1	DATA_DOMAIN	EXD			
2	REF_AREA	JP			
3	COUNTERPART_AREA	_Z			
4	FREQ	Q			
5	UNIT_MULT				
6	OBS_STATUS				
7					
8	INDICATOR	2014-Q1	2014-Q2	2014-Q3	2014-Q4
9					
10					

Figure 17 – showing a generated Excel template

The generated template may contain a header section, this section (cells A1:A6, B1:B6 in the image above) contains any dimensions which contain only one value for the entire dataset. This will occur if any reporting restrictions exist for the Dataflow, or Data Provider/Provision Agreement. In the image above, a number of Dimensions contain only one allowable value, and have therefore been pre-populated. The additional header cells (UNIT_MULT and OBS_STATS) which do not contain a value, are placeholders for Observation level attributes. These placeholders allow a default value to be set for all observations in the entire dataset.

Following the header section is the data section. This includes columns which have been created for all the additional Dimensions and Attributes, and time periods. In the image above, Indicator is the only remaining Dimension which does not have a fixed value. In the above example, the Time periods iterate over a quarterly frequency.

An example of a completed dataset is shown below.

	A	B	C	D
1	**DATA_DOMAIN**	EXD		
2	**REF_AREA**	JP		
3	**COUNTERPART_AREA**	_Z		
4	**FREQ**	Q		
5	**UNIT_MULT**	9		
6	**OBS_STATUS**	A		
7				
8	**INDICATOR**	**2014-Q1**	**2014-Q2**	**2014-Q3**
9	DG_GR_BP6_XDC	92,098	94,893	98,900
10	DG_GRS_BP6_XDC	50,625	52,330	49,869
11	DG_GRSD_BP6_XDC	50,323	52,036	49,549

Figure 18 showing an Excel Data Template, filled in with the series and observation data

5 CSV

5.1 Exporting CSV

Data can be output in three CSV formats: CSV Flat; CSV SDMX; CSV Time Series.

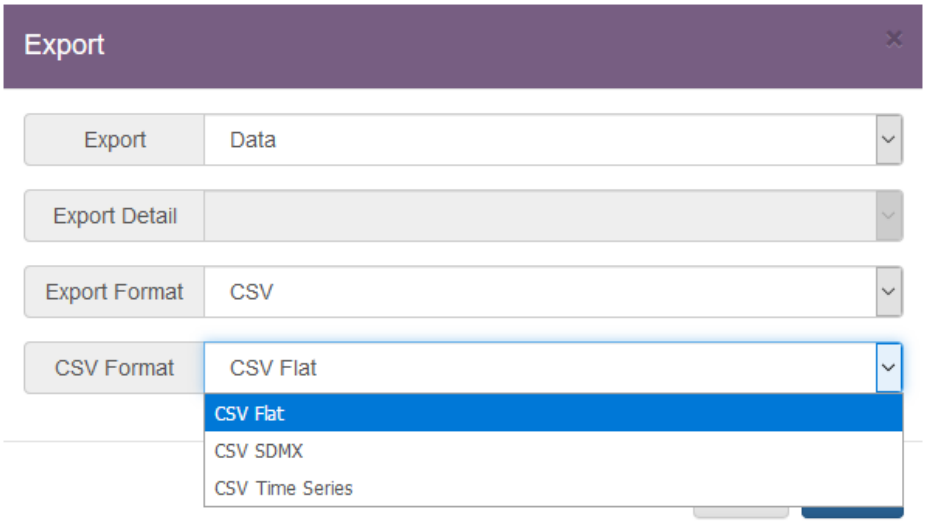


Figure 19 showing the CSV export formats

The CSV Flat format will output the data with one observation per row. A header row detailing what each column in the file represents. Subsequent rows will then have columns in the following order:

- Columns for each dimension
- Columns for each of the Dataset Attributes and Series Attributes
- A column for the Time Period
- A column for the measure
- Columns for each of the observation attribute.

All attributes defined in the DSD are present in the output, regardless of whether these attributes are specified within the Dataset. If the Registry has been configured to output Names as well as IDs, then for each coded column there will be an adjacent column stating the localised name. Figure 20 below shows an example output. The coded values are supplied in one column, with the localised name in the next (e.g. FREQ and Frequency). Uncoded dimensions (e.g. DECIMALS) are only listed once. Blank attributes (SERIESTITLE and SERIESALIAS) have no value.

	A	B	C	D	E	F	G	H	I	J	K
1	FREQ	Frequency	BASE_CUR	Base currency	DECIMALS	SERIESTITLE	SERIESALIAS	CALCULATED	FACTOR	TIME_PERIOD	OBS_VALUE
2	B	Business	NOK	Norwegian krone	4			FALSE	1	30/05/2014	5.5637
3	B	Business	NOK	Norwegian krone	4			FALSE	1	02/06/2014	5.5461
4	A	Annual	NOK	Norwegian krone	2			TRUE	1	2016	113.40
5	A	Annual	NOK	Norwegian krone	4			FALSE	1	2014	6.3019
6	A	Annual	NOK	Norwegian krone	5			FALSE	1	2015	8.07391

Figure 20 showing a CSV Flat output with IDs and Names

The CSV SDMX format is similar to the CSV Flat format with a single observation per row and a header row. However each row starts with the Dataflow column and that the code values and code names are output together in each column. The order of the columns is:

- Column for the Dataflow - in the form: AGENCY:DATAFLOW_ID(VERSION)

- Columns for each dimension
- A column for the Time Period
- A column for the measure
- Columns for each of the attributes (in order: Dataset, Series then Observation)

So for a coded column (such as FREQ), the values are output together within a single delimiter. Figure 21 below shows an example output.

	A	B	C	D	E	F	G
1	Dataflow	FREQ:Frequency	BASE_CUR:Base currency	TIME_PERIOD:Time period	OBS_VALUE:Observation value	DECIMALS:Decimals	SERIESTITLE:Series titl
2	NB:EXR(1.0): Exchange rates	B: Business	NOK: Norwegian krone	30/05/2014	5.5637	4	
3	NB:EXR(1.0): Exchange rates	B: Business	NOK: Norwegian krone	02/06/2014	5.5461	4	
4	NB:EXR(1.0): Exchange rates	A:Annual	NOK: Norwegian krone	2016	113.40	2	
5	NB:EXR(1.0): Exchange rates	A:Annual	NOK: Norwegian krone	2014	6.3019	4	
6	NB:EXR(1.0): Exchange rates	A:Annual	NOK: Norwegian krone	2015	8.07391	5	

Figure 21 showing CSV SDMX output

CSV Time Series format outputs the dimensions and all of the dates in the dataset in the first row. Subsequent rows list the series and all of the observations for that series. This format will generally result in a wide CSV file.

Note that attributes are not output in this format. Figure 22 below shows an example output.

	A	B	C	D	E	F	G	H	I	J	K
1	FREQUENCY	Frequency	BASE_CUR	Base currency	Series	30/05/2014	02/06/2014	2016	2014	2015	
2	B	Business	NOK	Norwegian krone	B:NOK	5.5637	5.5461				
3	A	Annual	NOK	Norwegian krone	A:NOK			113.4	6.3019	8.07391	

Figure 22 showing CSV Time Series output

All three of the CSV outputs will include quotation marks around a column value only if the value contains the delimiter.

5.2 Importing CSV

The Fusion Registry supports the use-case of importing a flat dataset (one observation per row) in two different formats.

5.2.1 Fusion Registry CSV Format

The data file may not have a header row. The columns for each row of the CSV file must correspond to the following order:

- Each of the dimensions in the Data Structure in their natural order
- The measure dimension
- The attributes for the Data Structure (in the order: DataSet, Series, Observation)
- The observation value

The delimiter can either be a tab, comma, semicolon, or whitespace.

The CSV snippet below shows 3 observations for the same series. The time periods are 2010, 2011, and 2012 respectively with observation values of 100.01, 102.03, and 105.05

```
01R,CA,A_IX,CA,A,2010,Comment,Base Per,0,P1Y,A,100.01
01R,CA,A_IX,CA,A,2011,Comment,Base Per,0,P1Y,A,102.03
01R,CA,A_IX,CA,A,2012,Comment,Base Per,0,P1Y,A,105.05
```

5.2.2 SDMX CSV Format

The data file may or may not have a header row. If there is a Header row this must start with the word “Dataflow”. The columns for each row of the CSV file must correspond to the following order:

- The dataflow in the format: <agency>:<id>(<version>)
- Each of the dimensions in the Data Structure in their natural order
- The measure dimension
- The observation value
- The attributes for the Data Structure (in the order DataSet, Series, Observation)

The delimiter can either be a tab, comma, semicolon, or whitespace.

The CSV snippet below shows a CSV with a header row and 2 observations for the same series. The time periods are 2010 and 2011 with observation values of 100.01 and 100.02.

```
Dataflow,DATA_DOMAIN,REF_AREA,INDICATOR,COUNTERPART_AREA,FREQ,TIME_PERIOD,O
BS_VALUE,COMMENT,BASE_PER,UNIT_MULT,TIME_FORMAT,OBS_STATUS
IMF:ECOFIN_DF(1.0),01R,CA,A_IX,CA,A,2010,100.01,Comment,Base Per,0,P1Y,A
IMF:ECOFIN_DF(1.0),01R,CA,A_IX,CA,A,2011,100.02,Comment,Base Per,0,P1Y,A
```