



FUSION TRANSFORMER

User Guide

FUSION TRANSFORMER

Fusion Transformer is a command-line application providing transformations between SDMX and GESMES data files and structure files.

User Guide

This document explains how to quickly get started with the Fusion Transformer and the formats it supports.

Table of Contents

1	Overview	2
2	Data Transformation	3
3	<i>Structure Transformation</i>	4
4	<i>File Formats and Character Encoding</i>	5
5	<i>Appendix 1 – Error Codes</i>	6
5.1	Data Transformer	6
5.2	Structure Transformer	6

1 Overview

The Fusion Transformer is a command line application providing transformations between SDMX, and GESMES/TS data files and structure files.

The following data file formats are supported:

- Generic 1.0
- Generic 2.0
- Generic 2.1
- GenericTimeSeries (input only)
- Compact 1.0
- Compact 2.0
- StructureSpecific 2.1
- StructureSpecificTimeSeries 2.1 (input only)
- Message Group 1.0 (containing Compact or Generic Datasets only)
- Message Group 2.0 (containing Compact or Generic Datasets only)
- GESMES/TS

The following structure file formats are supported:

- SDMX Structure Document 1.0
- SDMX Structure Document 2.0
- SDMX Structure Document 2.1
- SDMX Registry Interface Document 2.0
- GESMES/TS

2 Data Transformation

The Data Transformer can be run by executing the command:

```
java -cp FusionTransformer.jar com.metadatatechnology.fusion.dataparser.DataParserMain
```

The following additional arguments are available:

Argument	Mandatory	Description	Allowed Arguments
-d <arg>	False	URI of data file to transform. If this option is not specified or the argument is - then input is taken from Standard Input.	
-f <arg>	True	Output data format	compact / generic / edi
-o <arg>	False	Output file. If this option is not specified or the argument is - then output is sent to Standard Output.	
-s <arg>	True	URI of structure file with DSD	
-v <arg>	False	Output SDMX Version if appropriate	1.0 / 2.0 / 2.1
-split	False	Split multiple datasets into individual files	
-tf	False	Prevents the Time Format attribute from being output in SDMX-EDI	

For convenience there is a dataTransform.bat file provided that Windows users can use to launch the main class. Example usage:

```
dataTransform.bat -d TestData/inputData.ges -s TestData/inputDSD.ges -o output.xml -f compact
```

For UNIX users there is an equivalent file: dataTransform.sh.

3 Structure Transformation

The Structure Transformer can be run by executing the command:

```
java -cp SdmxTransformer.jar com.metadatatechnology.fusion.dataparser.StructureParseMain
```

The following additional arguments are available:

Argument	Mandatory	Description	Allowed Arguments
-s <arg>	False	URI of structure file to transform. If this option is not specified or the argument is - then input is taken from Standard Input.	
-o <arg>	False	Output file. If this option is not specified or the argument is - then output is sent to Standard Output.	
-v <arg>	True	The output version	edi / edi-lenient/ 1.0 / 2.0 / 2.1

There are 5 options for the output version. Supplying the argument 1.0, 2.0 or 2.1 will result in the creation of an SDMX-ML file in the specified format. Supplying the argument EDI will create an EDI file if possible.

The argument EDI-LENIENT will create an EDI file ignoring the error condition when a structure references another structure belonging to a different Agency and will return a file where structures are in the same Agency as required. For example, a Data Structure Definition in Agency 1 if it refers to a Codelist in Agency 2 cannot strictly be converted into EDI since EDI cannot express that the DSD and Codelist are in different agencies. By using EDI-LENIENT this restriction is ignored and the outputted EDI file would state that both are owned by Agency1.

For convenience there is a structureTransform.bat file provided that Windows user can use to launch the main class. Example usage:

```
structureTransform.bat -o ediStructreOut21.edi -s StructureOut21.xml -v edi
```

For UNIX users there is an equivalent file: structureTransform.sh.

4 *File Formats and Character Encoding*

The Fusion Transformer expects that all files supplied to it are encoded using the charset UTF-8, except for EDI files which are expected to be encoded using the charset ISO-8995-1. If you supply a file that is not explicitly encoded to the appropriate charset, the Fusion Transformer will make the assumption that it is encoded in the expected charset and attempt to decipher it using that charset.

All of the files generated by the Fusion Transformer will be encoded using the charset UTF-8, except for EDI files which will be encoded using the charset ISO-8995-1.

If you experience the Fusion Transformer behaving strangely with certain characters, please check the encoding of your input files using an appropriate software tool.

5 Appendix 1 – Error Codes

The following return codes are returned by the applications:

5.1 Data Transformer

Return Code	Meaning
0	Application completed successfully.
1	No arguments specified.
2	Illegal argument specified.
3	Data File could not be found.
4	Structure File could not be found.
5	Specified schema version incorrect.
6	Illegal output data format specified.
7	Cannot write to output file.
10	Error during transformation.

5.2 Structure Transformer

Return Code	Meaning
0	Application completed successfully.
1	No arguments specified.
2	Illegal argument specified.
4	Structure file could not be found.
6	Illegal output data format specified.
7	Cannot write to output file.
10	Error during transformation.