

Ili2fme

From Fmepedia

ili2fme (<http://www.eisenhutinformatik.ch/interlis/ili2fme>) is an Open Source FME Plugin for reading and writing data according to the Swiss Standard For Geodata Exchange "INTERLIS".

This page aims to describe how to give some help to use the plugin.

Table of contents

- 1 What it does
- 2 Where to find it
- 3 How to install it
- 4 Reading INTERLIS1 Data
- 5 Reading INTERLIS2 Data
- 6 Writing INTERLIS1 Data
- 7 Writing INTERLIS2 Data
- 8 List Handling
- 9 Examples

What it does

ili2fme can:

- Read and Write INTERLIS1-Data
- Read and Write INTERLIS2-Data

Where to find it

- In FME 2007, the version 3.1 of ili2fme is included in the standard version.
- More recent versions can be found here (<http://www.eisenhutinformatik.ch/interlis/ili2fme>).

How to install it

- To install the new version, the content of the folder "FME Suite" has to be copied to the FME_HOME directory (e.g. c:\program files\fme)
- To use the plugin, a Java Virtual Machine has to be installed on your PC.
- To install it on pre-FME2007 versions, the JVM.DLL has to be on the Path!
- To use ili2fme in the FME Viewer, the following Environment Setting has to be set:
FME_VIEWER_THREADING=SINGLE

Reading INTERLIS1 Data

To read INTERLIS1 Data, the Model (.ili) must be known to FME.

It can be stored:

- in %fme_home%\plugins\interlis2\ilimodels
- in a special model directory you specify
- in the same directory than your data

Then you can select an INTERLIS1-Data-File (.itf) and open it with FME (Viewer, Workbench, Universal Translator) and use it.

- All the enumerations from the ITFs will be converted to texts (values).
- If more than one geometry exists, the first geometry will be used as FME geometry, the other ones will be stored as Hex Well Known Binary in Attributes. Starting with Release 4.4.0, it is not the first geometry but the first geometry that is not a point that will be the main geometry and all the others will be stored in an attribute.

Reading INTERLIS2 Data

Reading INTERLIS2 - Data is essentially the same than reading INTERLIS1-Data with the following differences:

- The data comes in XTF - Files (and not ITF-Files)
- If your data-models contains EXTENDS, FME will show all the data in a single "superstructure" - Featuretype. You will have to use an AttributeFilter on XTF_CLASS to separate the different classes in Workbench. Since v. 4.4.0, the datamodel may also be imported with a "subclass"-strategy rather than a superclass strategy. When "subclass" is chosen a featuretype is created for each extended objects, whereas one featuretype is created per parent object when "superclass" is chosen.

Writing INTERLIS1 Data

To write INTERLIS1-Data, the process is the following

Prerequisites: the INTERLIS Model (.ili) has to exist before!

- Set up a Workbench
- Define an "INTERLIS (ili2fme) Writer"
- **Import**
the FeatureTypes from your ILI-Model (Destination Data -> Import FeatureTypes -> Browse to your ILI-File)
- As a table in INTERLIS can contain more than one geometry attribute, you have to define which geometry attribute to use. You may expose the format attribute called "xtf_geomattr" and set it to a constant value
- Define a transfer-ID "XTF_ID" (e.g. generate it with a counter or map a format attribute like OBJECTID / FID or similar)
- Route your features to this featuretype (connect the arrows)
- Define the MODEL-Name in the Settings of the "INTERLIS (ili2fme)" - Writer (on the lefthand - side)
- GO!

Attention, if you work with surfaces or areas, you will have to split them into separate tables (explanations will come ...) (see the example in gdb2ili.fmw in the attachment below)

Writing INTERLIS2 Data

To write out INTERLIS2-Data, you will have to follow these steps in addition to the ones explained for INTERLIS1:

- Create a basket for each TOPIC (With a Creator / NullGeometryCreator + AttributeCreator)
- Reference this basket in each featuretype of the topic "XTF_BASKET" (e.g. by attaching a constant)
- Write all herited classes to a "superstructure" featuretype. (or choose a subclass-strategy)
- Define the qualified classname of each class in "XTF_CLASS"

If you have trouble getting this to work, consider the following:

- Only XTF_Basket attributes must be created by hand in a common transformation with an Interlis writer.
- Depending on what format transformation one carries out, (e.g. ili->shp) the xtf_geomattr geometry attribute must be exposed and a constant (i.e. the geometry type name) must be attached to it. In other cases, such as ili->ili semantic transformations, the xtf_class format attribute needs to be exposed and filled with the qualified target-model class-name by hand.
- Perhaps it is only necessary to expose the xtf_geomattr when you carry out a XYZ->ili transformation (e.g. shp->ili) because the inner geometry model of shp is completely different from the one in Interlis (geometry = "normal", i.e. structured class attribute; classes may as well have no geometry at all...)
- You always need to provide fully classified names of the target model. For example, the correct parameter might be: "Beispiel_ILI2_Model.Features". Source model (reader): Beispiel_GINA_Model (new version: GINAModel) Target model (writer): Beispiel_ILI2_Model (new version: ILI2Model)

List Handling

In INTERLIS2, there can be lists, such as "BAG OF", "LIST OF", STRUCTUREREF.

In this case, an FME - list is given to every INTERLIS-feature. In the example below, there is an illustration about how to work with these lists.

Examples

There are two examples here:

A General one illustrating how to

- import data from an ITF to an ESRI Geodatabase (MDB)
- export data from the ESRI Geodatabase to an ITF

The data is the freely available testdataset from swisstopo's VECTOR25. The Geodatabase schema has been generated with iliX (<http://ilix.inser.ch>).

List-Handling

- This example show how to work with lists in INTERLIS2

Retrieved from "<http://www.fmepedia.com/index.php/Ili2fme>"

Categories: FAQ | Interlis Format FAQ

Attached Files

file	size	date
ili2fmeListSample.zip	7.4 kB	12/12/07
ili2fme_samples.zip	568.5 kB	12/12/07

User Comments Add a new comment

- This page was last modified 21:06, 10 Jan 2008.