

Problem Solving with the Interlis 2.0 Writer

From Fmepedia

This Q and A was taken from an actual Safe Support email discussion and is intended to provide a "knowledge base" for xyz->ili, ili->xyz and ili->ili transformation appliers in order to help others who might have similar challenges.

Table of contents

- 1 Q) In some cases the log appeared to complete successfully, but then when I tried to open the output.xtf all FME viewer could read was the single xtf_basket feature.
- 2 Q) What format attributes need to be populated? I exposed all the xtf_* format attributes on the reader and writer feature types to make sure they were being populated (xtf_basket, xtf_class, xtf_geomattr).
- 3 Q) I added xtf_geomtype to all the destination classes and set it using a GeometryFilter and AttributeCreator.
- 4 Q) Is the XTF_Basket feature supposed to have an xtf_topic value of 'Features' or 'Beispiel_ILI2_Model.Features' or 'Beispiel_GINA_Model.Features'?
- 5 Q) Why do we need to write to feature types = Beispiel_ILI2_Model.Features.polnum etc and not Beispiel_GINA_Model.Features.polnum etc?
- 6 Q) Why doesn't a straight across translation work? That is, read from input.xtf and write to output.xtf with all feature types mapped directly across XTF_BASKETS to XTF_BASKETS ModelA.TopicA.ClassC to ModelA.TopicA.ClassC etc?
- 7 Q) What is the purpose of the PointOnArea overlay given that none of the output geometry types are supposed to be areas?
- 8 Q) Are the xtf_class names supposed to be the same as the source fme_feature_type or does the 'Beispiel_GINA_Model.Features' need to be changed to 'Beispiel_ILI2_Model.Features'?
- 9 Q) Do both models need to be referenced in the writer parameter: 'Models'? Currently only Beispiel_ILI2_Model is referenced, but not Beispiel_GITA_Model.
- 10 Q) Did we import the schemas from the destination model directly, or should we generate the destination feature types based on the input schemas?
- 11 Q) Why don't we have any destination geom feature types? I tried adding Beispiel_ILI2_Model.Features.gebaeude_geom and Beispiel_ILI2_Model.Features.bauzone_geom but was not able to get the translation to complete.
- 12 Q) Overall, is getting this to work is largely a configuration issue?

Q) In some cases the log appeared to complete successfully, but then when I tried to open the output.xtf all FME viewer could read was the single xtf_basket feature.

A) In the end, that's what FME did actually right: not finishing the transformation until all input features are correct...

Q) What format attributes need to be populated? I exposed all the xtf_ format attributes on the reader and writer feature types to make sure they were being populated (xtf_basket, xtf_class, xtf_geomattr).*

A) Actually, according the manual, 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.

Q) I added xtf_geomtype to all the destination classes and set it using a GeometryFilter and AttributeCreator.

A) 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...).

Q) Is the XTF_Basket feature supposed to have an xtf_topic value of 'Features' or 'Beispiel_ILI2_Model.Features' or 'Beispiel_GINA_Model.Features'?

A) You always need to provide fully classified names of the target model. In this example, the correct parameter would be: "Beispiel_ILI2_Model.Features". In the modified model, it would be "ILI2Model.Features", respectively. Source model (reader): Beispiel_GINA_Model (new version: GINAModel) Target model (writer): Beispiel_ILI2_Model (new version: ILI2Model)

Q) Why do we need to write to feature types = Beispiel_ILI2_Model.Features.polnum etc and not Beispiel_GINA_Model.Features.polnum etc?

A. Because we have to write to the target model which is called "Beispiel_ILI2_Model.whatever" or "ILI2Model.whatever" in the new version.

Q) Why doesn't a straight across translation work? That is, read from input.xtf and write to output.xtf with all feature types mapped directly across XTF_BASKETS to XTF_BASKETS ModelA.TopicA.ClassC to ModelA.TopicA.ClassC etc?

A. It should work if the source data is ok. You can try a new tool from Claude's, which is particularly designed for Interlis 2 datasets and models. This new (beta-status) tool can check the references correctly and help you find out what is wrong with the data...

Q) What is the purpose of the PointOnArea overlay given that none of the output geometry types are supposed to be areas?

A. Input features consist of a set of polylines that are to be transformed into area tessellation borders. The point features are actually the centroids of these areas and carry all thematic information. So, the PointOnArea overlay combines the area borders with the appropriate thematic information from the former centroid features.

[Remark: if you use an Interlis 1 writer which produces a data file in the old Interlis 1 format, you actually would have needed the centroid coordinates at the end because the "old" tessellation encoding consists of a table with the area borders and another table with the appropriate centroid coordinates carrying all thematic information. The relation between borders and centroids would in this case not be explicit, but geometric.]

Q) Are the xtf_class names supposed to be the same as the source fine_feature_type or does the 'Beispiel_GINA_Model.Features' need to be changed to 'Beispiel_ILI2_Model.Features'?

A. Yes, the names need to be changed according to the target data model.

Q) Do both models need to be referenced in the writer parameter: 'Models'? Currently only Beispiel_ILI2_Model is referenced, but not Beispiel_GITA_Model.

A. In ili2fine, the source model-name is read from the dataset. Each feature element starts with the qualified class name. Hence, the data model must have the same name.

Q) Did we import the schemas from the destination model directly, or should we generate the destination feature types based on the input schemas?

A. Import the target schema via "Destination... -> Import Feature Type Definition"-dialog and generate the feature type from the input schema.

Q) Why don't we have any destination geom feature types? I tried adding Beispiel_ILI2_Model.Features.gebaeude_geom and

Beispiel_ILI2_Model.Features.bauzone_geom but was not able to get the translation to complete.

A) In an Interlis model, the geometry is basically a normal feature attribute. The two classes you mentioned above are normal classes as the others. They are not 'geometry types' as is. So, any Interlis class may or may not carry geometry type attributes (in the models the keywords are: COORD, POLYLINE, SURFACE and AREA).

Q) Overall, is getting this to work is largely a configuration issue?

A. Well, after all it depends much on input feature quality... But you need to know two or three "tricks" to get an Interlis translation successfully realized.

One suggestion - if you use AttributeCopiers and AttributeCreators to set things like xtf_class, then you dont need to reconnect them every time you edit your workspace - the xtf_class will stay green meaning it will stay populated without having to manually connect the fields again.

Retrieved from "http://www.fmepedia.com/index.php/Problem_Solving_with_the_Interlis_2.0_Writer"

Categories: [FAQ](#) | [Interlis Format FAQ](#)

Attached Files

file	size	date
gina2ili.pdf	77.3 kB	12/12/07
gina2ili.zip	103.2 kB	12/12/07

User Comments [Add a new comment](#)

- This page was last modified 20:53, 12 Dec 2007.