Complex GML

2014
FME
SAFE SOFTWARE
Lizard Island
Cloud Inlet
Transform Bay
Sea of Data
FME
F M E
2014

Complex GML - INSPIRE Focus

Contents

Introduction to Complex GML & INSPIRE

Many forms of complex GML exist. For the purposes of this training session the focus will be on primarily on INSPIRE GML, though some examples will also use AIXM5 GML. While these exercises focus on INSPIRE and AIXM schemas, the lessons and FME workflows illustrated here apply equally the many other flavours of complex GML exist.

INSPIRE, or Infrastructure for Spatial Information in Europe, is a European Union directive that aims to create a spatial data infrastructure of EU data, which would be used for policy making across boundaries, especially environmental. The directive also aims to improve public access to spatial information. For more information please visit the INSPIRE website. The INSPIRE Directive has data specifications that must be followed by EU member countries when submitting data, including providing data in INSPIRE compliant GML format.

INSPIRE data is divided into three Annexes. Each Annex contains a number of themes, which are groupings of spatial data. Some examples of themes include Geographical Names, Cadastral Parcels and Transport Networks. Each Annex has a different due date, so that the data can be submitted to the directive in a gradual fashion, rather than requiring countries to submit all their data at one time. Additionally, certain themes in Annex I are considered base themes, as some of their attributes are included in other themes, such as the Geographical Names. For example, the place name attribute is originally part of the Geographical Names theme, but is also included in the Protected Sites theme.

FME is a great tool to leverage complex GML. It can be used to achieve INSPIRE compliance, as FME can evaluate, assemble, transform, write, validate and publish INSPIRE compliant data.

FME offers a number of transformers and capabilities which can be used when working with INSPIRE GML. These include an INSPIRE GML Reader and Writer, web services capabilities, XML validation, format translation and schema mapping. New to FME 2014 is the INSPIRE GML Writer, which makes it easier to write INSPIRE GML. The generic GML writer also has a new application schema mode that allows FME users to publish to virtually any valid application schema they wish. In short, if FME can read it, then FME can write it.

Enhancements are continuing. With FME 2014-SP1, the GML Pretty Printing option makes it easier to read GML in a text editor. In FME 2014 - SP2 the INSPIRE schemas were updated to the latest versions to better support Annex II and III themes. To learn more about INSPIRE Reading and Writing, please review this article.

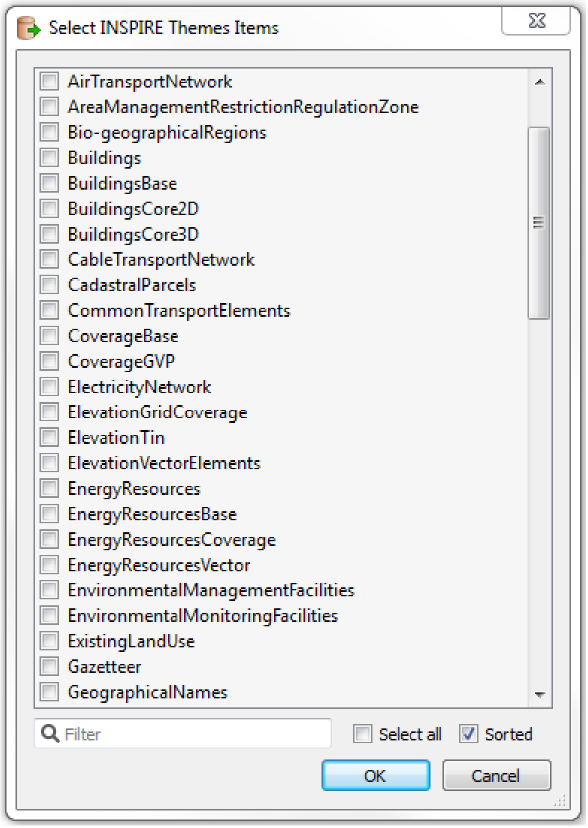
1. Reading INSPIRE Data

The INSPIRE samples can be read with the INSPIRE or GML readers. If you use the GML reader you may have to use the app schemas to read it depending on the namespace declarations and default namespace of the file you are reading. A well-constructed namespace header will provide valid links to the XML Schema Documents (XSDs) needed. When opening a file with Data Inspector, selecting no theme should work, and result in a fast response from the Data Inspector. However, it is also possible to choose the theme of interest. Note that the AIXM and INSPIRE schemas are delivered with FME in: <FME Home>\xml\schemas.

Exercise 1:

For this demo, INSPIRE GML will be re ad with the Data Inspector using the INSPIRE GML Reader. It is possible to select INSPIRE feature types by themes, or XSDs. The focus of this exercise will be cadastral parcels data from the UK.

1. Open Data Inspector and select open a dataset. For format select INSPIRE GML. To view sample datasets, open the source folder located at C:\FMEData2014\Resources\INSPIRE\ComplexGML-INSPIREfocus\source. It is possible to simply open and inspect any of the samples you are interested in.
2. A good example of a custom INSPIRE solution is Cadastral Parcels from the City of London.
   a. Try reading LandRegistry_CadastralParcels_CityofLondon.gml dataset using the INSPIRE reader set to the Cadastral Parcels theme.



Select INSPIRE Themes Items

□ AirTransportNetwork
□ AreaManagementRestrictionRegulationZone
□ Bio-geographicalRegions
□ Buildings
□ BuildingsBase
□ BuildingsCore2D
□ BuildingsCore3D
□ CableTransportNetwork
□ CadastralParcels
□ CommonTransportElements
□ CoverageBase
□ CoverageGVP
□ ElectricityNetwork
□ ElevationGridCoverage
□ ElevationTin
□ ElevationVectorElements
□ EnergyResources
□ EnergyResourcesBase
□ EnergyResourcesCoverage
□ EnergyResourcesVector
□ EnvironmentalManagementFacilities
□ EnvironmentalMonitoringFacilities
□ ExistingLandUse
□ Gazetteer
□ GeographicalNames

Filter □ Select all ☑ Sorted

OK Cancel

In Data Inspector, deselect all themes when opening an INSPIRE GML file

Reader Format: INSPIRE GML

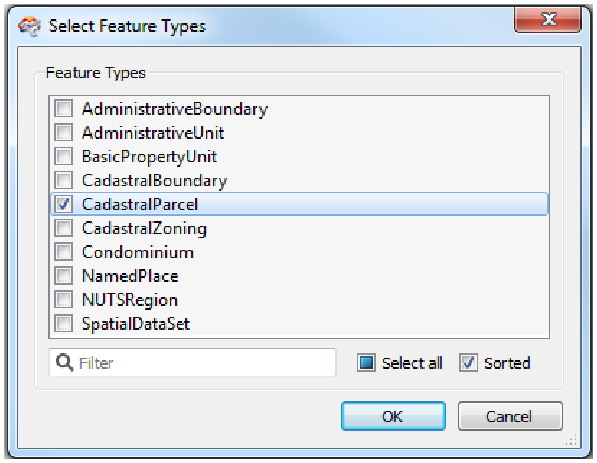
Reader Dataset:

```
C:\FMEData2014\Resources\INSPIRE\ComplexGML-
INSPIREfocus\source\UK.LandRegistry\LandRegistry_CadastralParcels_CityofLondon.gml
```

Reader Parameters

Select Feature Types: by Themes

INSPIRE Themes: CadastralParcels



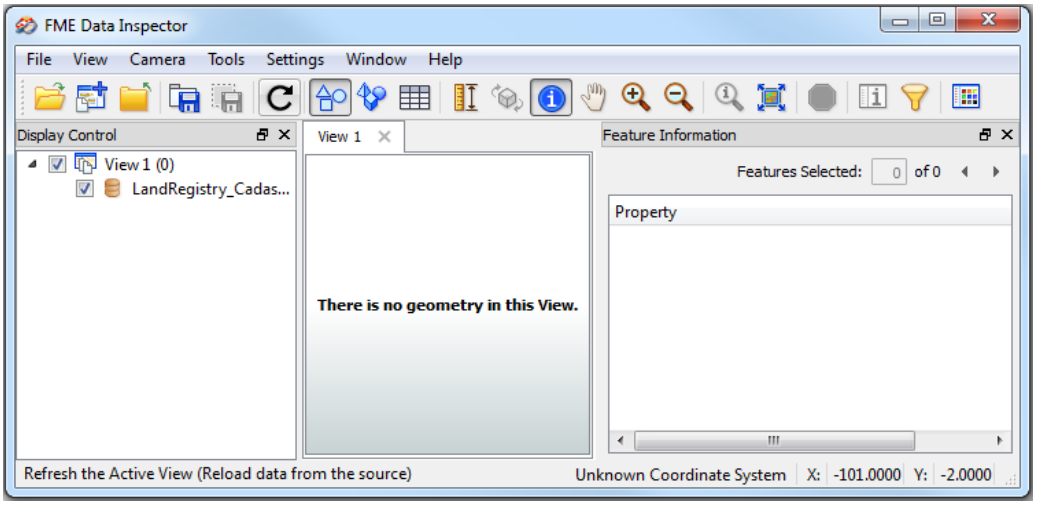
Select Feature Types

Feature Types

- □ AdministrativeBoundary
- □ AdministrativeUnit
- □ BasicPropertyUnit
- □ CadastralBoundary
- ☑ CadastralParcel
- □ CadastralZoning
- □ Condominium
- □ NamedPlace
- □ NUTSRegion
- □ SpatialDataSet

Filter ▪ Select all ☑ Sorted

Question: What happens in Data Inspector?



☑ View 1 (0)

☑ LandRegistry_Cadas...

There is no geometry in this View.

Features Selected: 0 of 0

Property

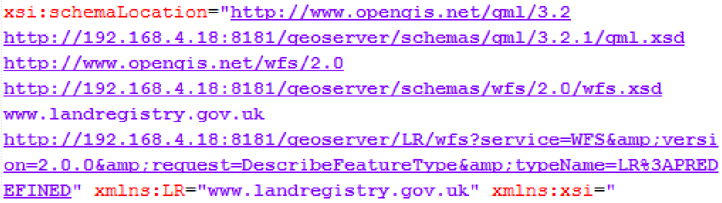
Refresh the Active View (Reload data from the source)

Unknown Coordinate System X: -101.0000 Y: -2.0000

Answer: Data is not read and there is no geometry to view with Data Inspector. We will look into this problem in the following steps.

b. Open LandRegistry_CadastralParcels_CityofLondon.gml in an XML or text editor. Save as XML, so that it is possible to view as pretty print XML.

Question: What potential problem(s) do you see with it?



```
xsi:schemaLocation="http://www.opengis.net/gml/3.2
http://192.168.4.18:8181/geoserver/schemas/gml/3.2.1/gml.xsd
http://www.opengis.net/wfs/2.0
http://192.168.4.18:8181/geoserver/schemas/wfs/2.0/wfs.xsd
www.landregistry.gov.uk
http://192.168.4.18:8181/geoserver/LR/wfs?service=WFS&amp;versi
on=2.0.0&amp;request=DescribeFeatureType&amp;typeName=LR%3APRED
EFINED" xmlns:LR="www.landregistry.gov.uk" xmlns:xsi="
```

Schema location for London cadastral parcels

Answer: The schema is local, which explains why we were unable to open this data using INSPIRE themes. For the file to be viewed in Data Inspector, it requires a custom XSD to be specified.

c. Back in Data Inspector, select open a dataset. For format select INSPIRE GML again, but now set the application schema parameter to point to landregistryuk.xsd.

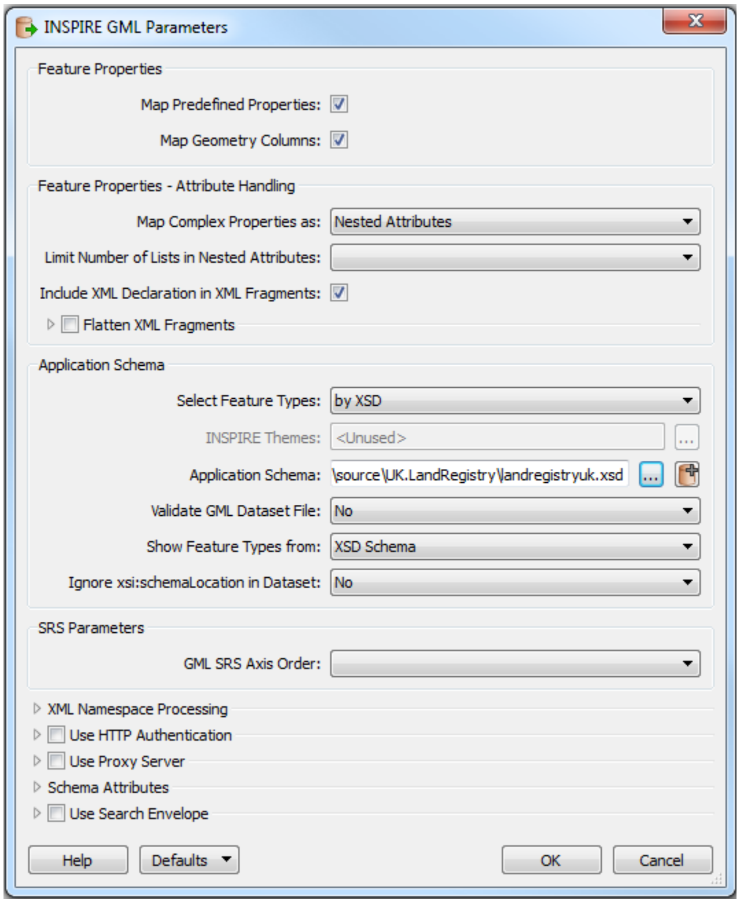
Reader Parameters

Select Feature Types: by XSD

Application Schema: C:\FMEData2014\Resources\INSPIRE\ComplexGML-INSPIREfocus\source\UK.LandRegistry\landregistryuk.xsd

Question: Now what happens in Data Inspector?

Answer: Cadastral parcels are displayed in Data Inspector. Using a custom version of the schema seems to defeat the purpose of using INSPIRE, yet as this situation may be encountered in the real world, we must be prepared to deal with it.



INSPIRE GML Parameters

Feature Properties

Map Predefined Properties: ☑

Map Geometry Columns: ☑

Feature Properties - Attribute Handling

Map Complex Properties as: Nested Attributes

Limit Number of Lists in Nested Attributes:

Include XML Declaration in XML Fragments: ☑

□ Flatten XML Fragments

Application Schema

Select Feature Types: by XSD

INSPIRE Themes: <Unused>

Application Schema: \source\UK.LandRegistry\landregistryuk.xsd

Validate GML Dataset File: No

Show Feature Types from: XSD Schema

Ignore xsi:schemaLocation in Dataset: No

SRS Parameters

GML SRS Axis Order:

XML Namespace Processing

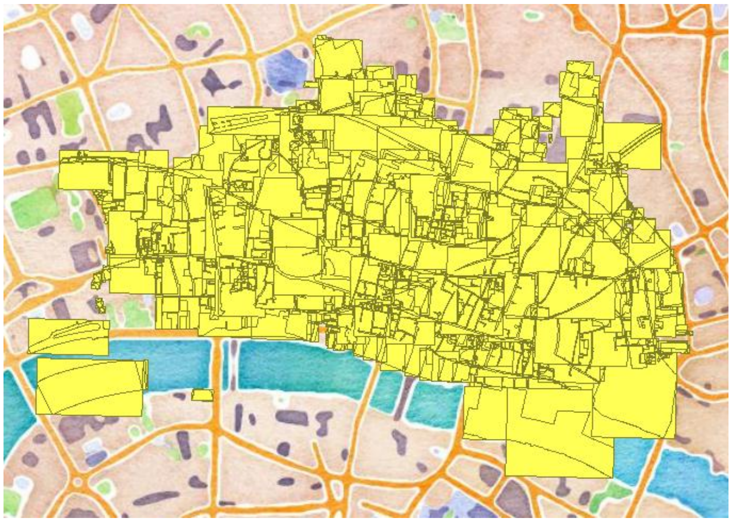
□ Use HTTP Authentication

□ Use Proxy Server

Schema Attributes

□ Use Search Envelope

GML reading using a local schema file



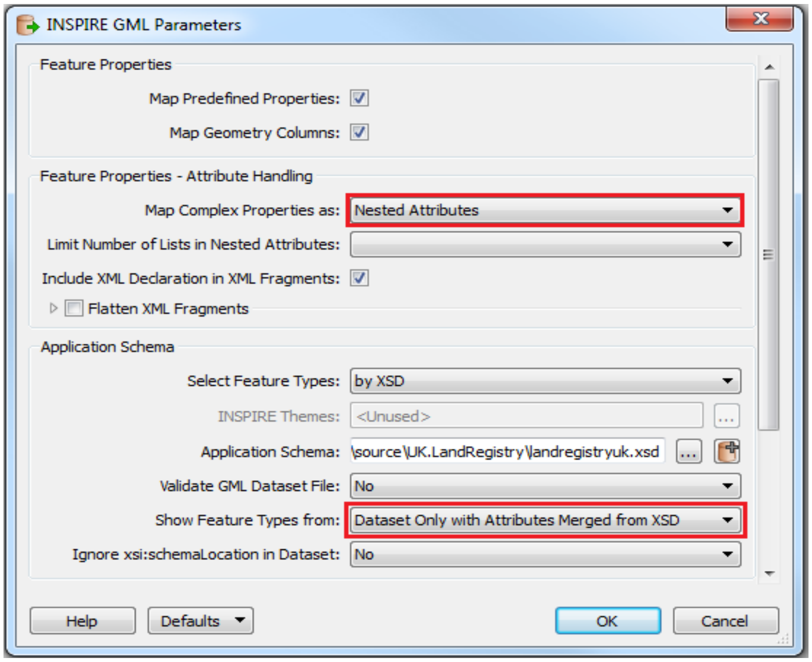
Land Registry UK INSPIRE Index Polygons

http://www.landregistry.gov.uk/professional/market-trend-data/inspire/inspire-downloads

http://data.inspire.landregistry.gov.uk/City_of_London.zip

3. Try reading any of the sample data sets with each of these 2 approaches to see the difference, using the INSPIRE GML Reader in Data Inspector. In general, it is good to use: 'Show feature types from: Dataset only with attributes merged from schema'. Note: Do not use flattening if you want to write out as XML or preserve the XML structure.

a. **Option A**: Map complex properties as: Nested Attributes



INSPIRE GML Parameters

Feature Properties

Map Predefined Properties: ☑

Map Geometry Columns: ☑

Feature Properties - Attribute Handling

Map Complex Properties as: Nested Attributes

Limit Number of Lists in Nested Attributes:

Include XML Declaration in XML Fragments: ☑

□ Flatten XML Fragments

Application Schema

Select Feature Types: by XSD

INSPIRE Themes: <Unused>

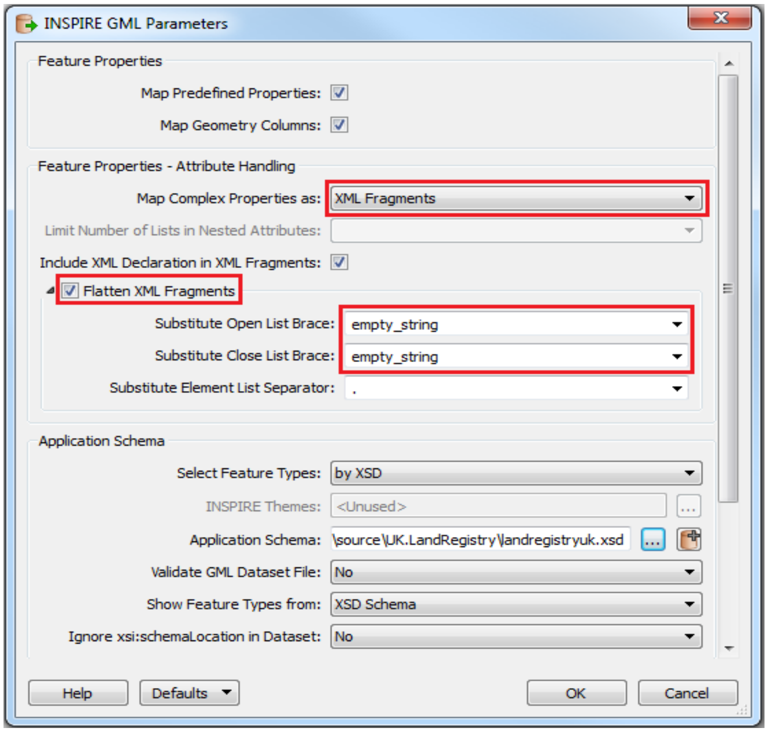
Application Schema: \source\UK.LandRegistry\landregistryuk.xsd

Validate GML Dataset File: No

Show Feature Types from: Dataset Only with Attributes Merged from XSD

Ignore xsi:schemaLocation in Dataset: No

b. **Option B**: Map complex properties as: XML fragments, check Flatten XML Fragments. Substitute Open List Brace "{" with empty_string, and substitute "}" with empty_string.



INSPIRE GML Parameters

Feature Properties

Map Predefined Properties: ☑

Map Geometry Columns: ☑

Feature Properties - Attribute Handling

Map Complex Properties as: XML Fragments

Limit Number of Lists in Nested Attributes:

Include XML Declaration in XML Fragments: ☑

☑ Flatten XML Fragments

Substitute Open List Brace: empty_string

Substitute Close List Brace: empty_string

Substitute Element List Separator: .

Application Schema

Select Feature Types: by XSD

INSPIRE Themes: <Unused>

Application Schema: \source\UK.LandRegistry\landregistryuk.xsd

Validate GML Dataset File: No

Show Feature Types from: XSD Schema

Ignore xsi:schemaLocation in Dataset: No

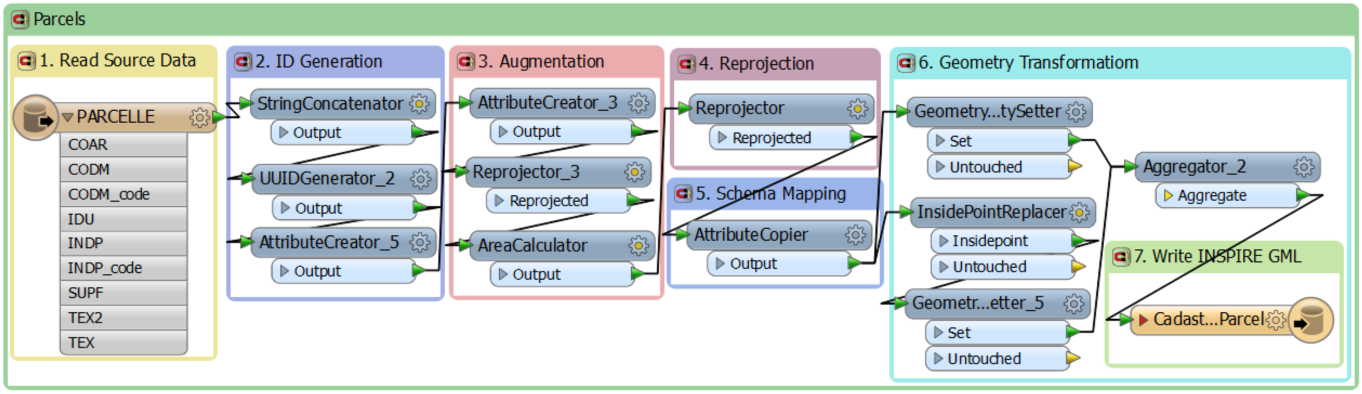
GML Reader settings for XML flattening

2. Writing INSPIRE GML

With FME, it is possible to obtain INSPIRE compliant GML within one small workspace. To write INSPIRE GML, the next steps are usually followed.

Typical INSPIRE Workflow:

1. **Read Source Data**. Source data is read in with the appropriate Reader.
2. **ID Generation**. IDs are generated, or if unique identifiers exist in the data, those are formatted as required.
3. **Augmentation of required fields**. Often source data is lacking certain attribute or attribute values required by the INSPIRE data specifications. These attributes, for example date and area info can be hard-coded, generated or obtained from schema mapping.
4. **Reprojection**. It is necessary to reproject the data to the official coordinate system.
5. **Schema Mapping**. Schema Mapping is performed where the source attributes are mapped to the correct destination attributes.
6. **Geometry transformation**. Geometry transformation consists of naming the geometry to match that of the xml_geometry type in the writer's feature type properties. Geometry transformation may also consist of combining or simplifying geometries.
7. **Write INSPIRE GML**. The XML schema document (XSD) for the writer can be specified either by selecting the INSPIRE theme or by specifying the file path for the XSD. Source attributes with the same name in the destination can be mapped directly to the destination feature type fields.



Cadastral parcels workflow, which demonstrates the usual steps to obtain INSPIRE GML

Exercise 2:

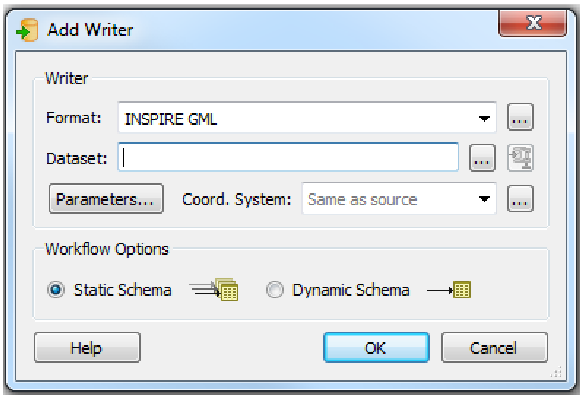
This example demonstrates INSPIRE GML writing. The workspace reads French national cadastre data and writes out INSPIRE GML according to the INSPIRE Annex I Cadastral Parcels schema (XSD). The output can be verified by reading it with the

INSPIRE GML Reader or validating it against the XSDs and XML syntax. The process outlined in the demo to write INSPIRE compliant GML can be applied to other INSPIRE themes. Section A uses an INSPIRE Writer, while Section B uses a GML Writer to obtain the output.

Section A: INSPIRE Writer

1. Open C:\FMEData2014\Resources\INSPIRE\ComplexGML-INSPIREfocus\INSPIRE.CadastralParcels\INSPIRE_CadastralParcels-Start.fmw.

2. Run workspace and inspect output.



Add Writer

Writer

Format: INSPIRE GML

Dataset:

Parameters... Coord. System: Same as source

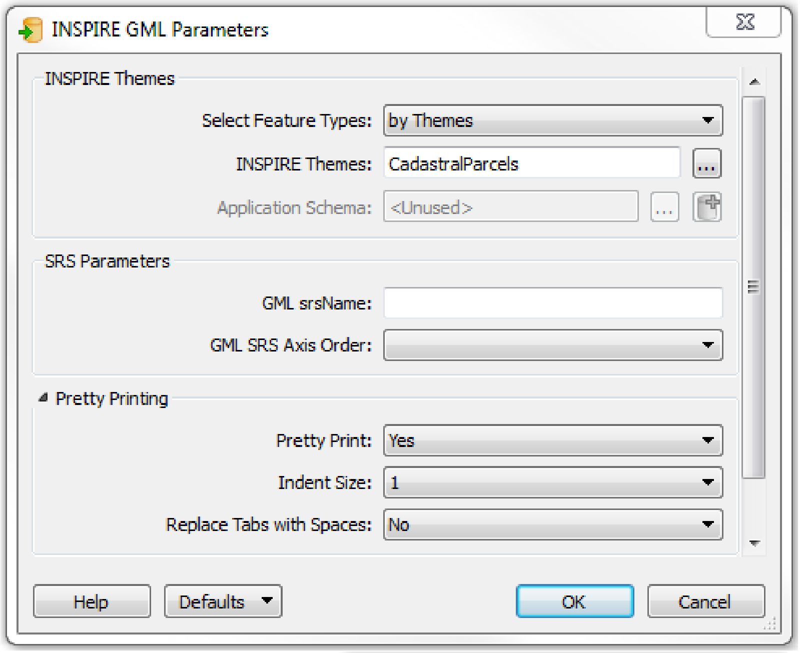
Workflow Options

● Static Schema ○ Dynamic Schema

3. Add INSPIRE Writers for both feature types in the workspace. It is possible to add an INSPIRE Writer by XSD Application Schema, or INSPIRE Theme.

    a. Select Add Writer (either by clicking on the Add Writer icon, or by choosing Writers > Add Writer). Enter INSPIRE GML as the format. Do not enter a name for the dataset at this time.

    b. Under Parameters, select feature types by themes and choose the INSPIRE theme 'CadastralParcels'. Select yes for the Pretty Print option.



INSPIRE GML Parameters

INSPIRE Themes

Select Feature Types: by Themes

INSPIRE Themes: CadastralParcels

Application Schema: <Unused>

SRS Parameters

GML srsName:

GML SRS Axis Order:

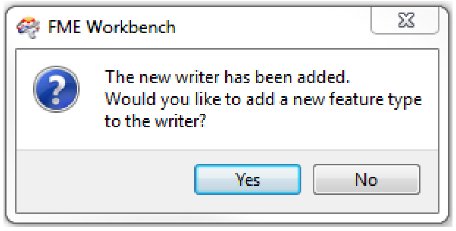
Pretty Printing

Pretty Print: Yes

Indent Size: 1

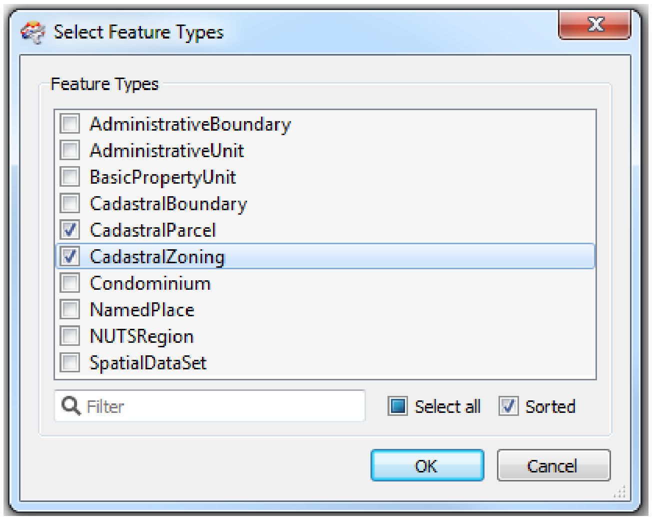
Replace Tabs with Spaces: No

Select No when asked if you would like to add a new feature type.



FME Workbench

The new writer has been added. Would you like to add a new feature type to the writer?



Select Feature Types

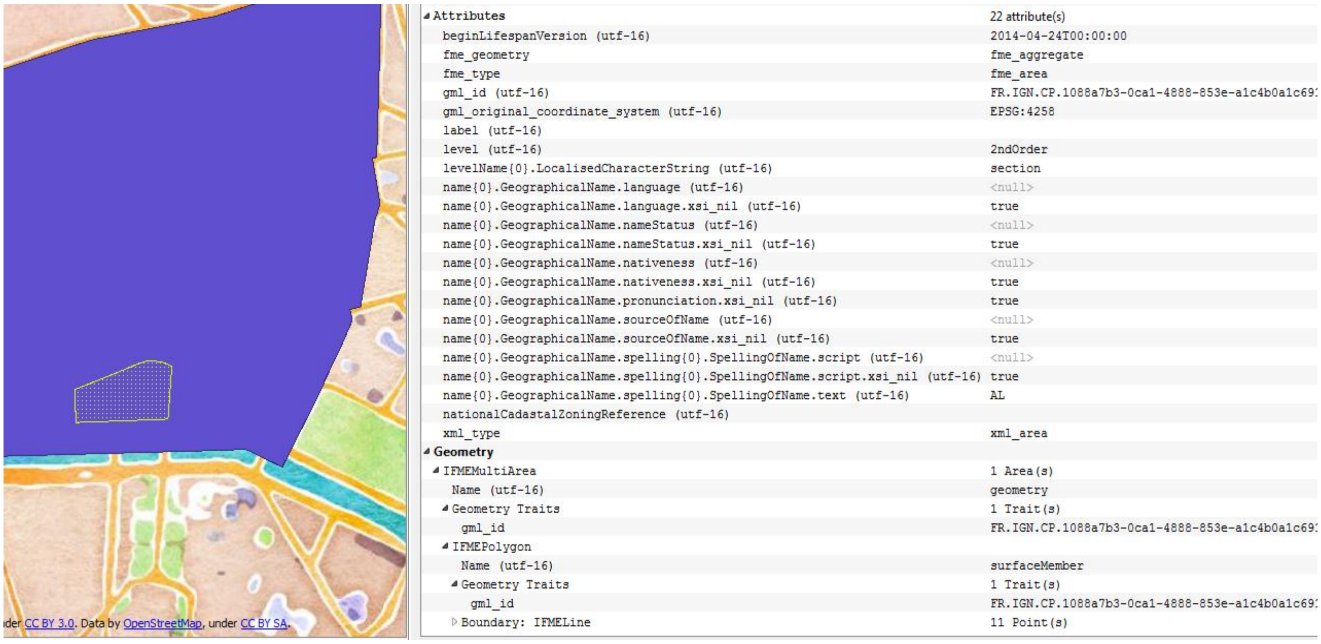
Feature Types

- ☐ AdministrativeBoundary
- ☐ AdministrativeUnit
- ☐ BasicPropertyUnit
- ☐ CadastralBoundary
- ☑ CadastralParcel
- ☑ CadastralZoning
- ☐ Condominium
- ☐ NamedPlace
- ☐ NUTSRegion
- ☐ SpatialDataSet

Filter ☐ Select all ☑ Sorted

4. Add features. Go to Writers - Import Feature Types.
    a. Under Parameters, select feature types by themes again and select the CadastralParcels theme. Push OK three times.
    b. Choose CadastralParcel and CadastralZoning feature types. Click OK.
    c. Connect outputs to the appropriate feature type instead of the inspectors.
5. Set output path.
6. Run workspace and inspect output using INSPIRE reader Workspace will be starting to look like INSPIRE_CadastralParcels-Complete.fmw, and results similar to output\Cadastral_Parcels.xml

Question: What's wrong with the output? How do we fix it?



```
◢ Attributes                                                            22 attribute(s)
  beginLifespanVersion (utf-16)                                         2014-04-24T00:00:00
  fme_geometry                                                          fme_aggregate
  fme_type                                                              fme_area
  gml_id (utf-16)                                                       FR.IGN.CP.1088a7b3-0ca1-4888-853e-a1c4b0a1c69
  gml_original_coordinate_system (utf-16)                               EPSG:4258
  label (utf-16)
  level (utf-16)                                                        2ndOrder
  levelName{0}.LocalisedCharacterString (utf-16)                        section
  name{0}.GeographicalName.language (utf-16)                            <null>
  name{0}.GeographicalName.language.xsi_nil (utf-16)                    true
  name{0}.GeographicalName.nameStatus (utf-16)                          <null>
  name{0}.GeographicalName.nameStatus.xsi_nil (utf-16)                  true
  name{0}.GeographicalName.nativeness (utf-16)                          <null>
  name{0}.GeographicalName.nativeness.xsi_nil (utf-16)                  true
  name{0}.GeographicalName.pronunciation.xsi_nil (utf-16)               true
  name{0}.GeographicalName.sourceOfName (utf-16)                        <null>
  name{0}.GeographicalName.sourceOfName.xsi_nil (utf-16)                true
  name{0}.GeographicalName.spelling{0}.SpellingOfName.script (utf-16)   <null>
  name{0}.GeographicalName.spelling{0}.SpellingOfName.script.xsi_nil (utf-16) true
  name{0}.GeographicalName.spelling{0}.SpellingOfName.text (utf-16)     AL
  nationalCadastalZoningReference (utf-16)
  xml_type                                                              xml_area
◢ Geometry
 ◢ IFMEMultiArea                                                        1 Area(s)
   Name (utf-16)                                                        geometry
  ◢ Geometry Traits                                                     1 Trait(s)
     gml_id                                                             FR.IGN.CP.1088a7b3-0ca1-4888-853e-a1c4b0a1c69
  ◢ IFMEPolygon
     Name (utf-16)                                                      surfaceMember
   ◢ Geometry Traits                                                    1 Trait(s)
       gml_id                                                           FR.IGN.CP.1088a7b3-0ca1-4888-853e-a1c4b0a1c69
   ▷ Boundary: IFMELine                                                 11 Point(s)
```

Data Source: Direction générale des Finances Publiques - Cadastre; Updated: 2013

Answer: Geometry is not output correctly. Multi-geometry needs to be assembled as required by INSPIRE cadastral parcels data specifications. The cadastral zoning feature has two geometry names, 'surfaceMember' for the inner geometry, the cadastral sections, and 'geometry' for the outer geometry, the municipality. Cadastral parcels also have two geometry names, 'geometry' and 'referencePoint'.

The ‘geometry’ name is for the parcel geometry, while the ‘referencePoint’ name is for the point geometry inside the parcel, for example the centroid of the cadastral parcel geometry. Geometries can be combined using the Aggregator to create multi-part geometry as per INSPIRE geometry requirements.

Section B: GML Writer

1. Open INSPIRE_CadastralParcels-Start.fmw.

2. Run workspace and show output.

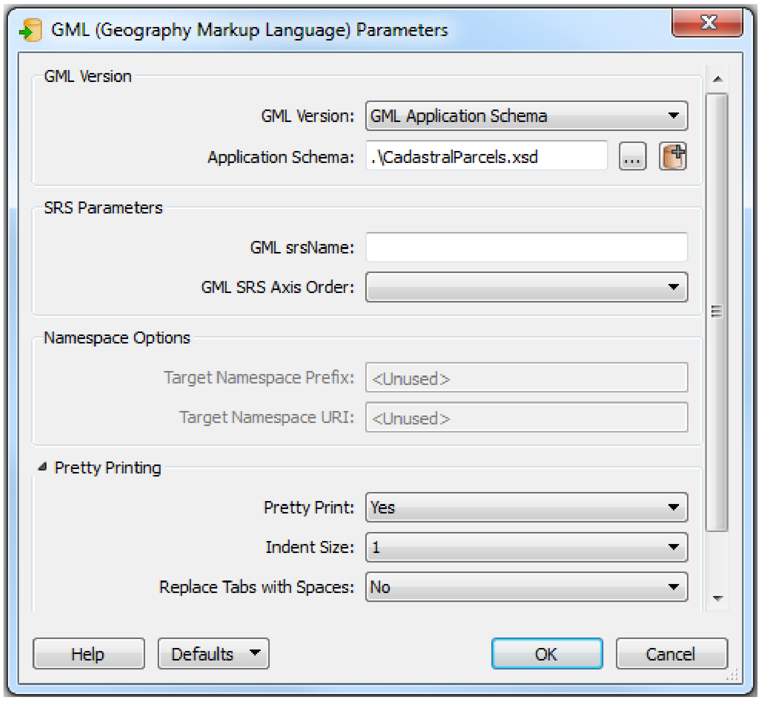
3. Add Writer.

Format: GML (Geography Markup Language)

Parameters:

GML Version: GML Application Schema

Pretty Printing: Yes



GML Version: GML Application Schema

Application Schema: .\CadastralParcels.xsd

GML srsName:

GML SRS Axis Order:

Target Namespace Prefix: <Unused>

Target Namespace URI: <Unused>

Pretty Print: Yes

Indent Size: 1

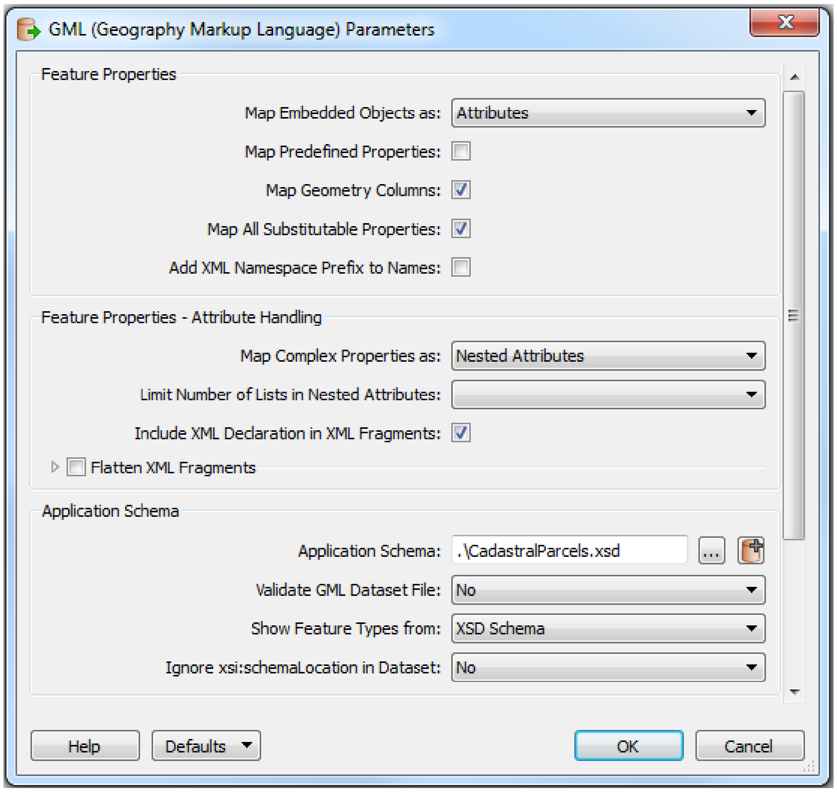
Replace Tabs with Spaces: No

GML Writer Parameters – Add dataset: New GML Application Schema mode

Application Schema: `C:\FMEData2014\Resources\INSPIRE\ComplexGML-INSPIREfocus\schemas\CadastralParcels.xsd`

Click OK. Select No to add new feature type.

4. Go to Writer - Import feature types. Go to reader parameters and set application schema to full path for 'CadastralParcels.xsd', and click ok twice.


GML (Geography Markup Language) Parameters

Feature Properties

Map Embedded Objects as: Attributes

Map Predefined Properties: □

Map Geometry Columns: ☑

Map All Substitutable Properties: ☑

Add XML Namespace Prefix to Names: □

Feature Properties - Attribute Handling

Map Complex Properties as: Nested Attributes

Limit Number of Lists in Nested Attributes:

Include XML Declaration in XML Fragments: ☑

□ Flatten XML Fragments

Application Schema

Application Schema: .\CadastralParcels.xsd

Validate GML Dataset File: No

Show Feature Types from: XSD Schema

Ignore xsi:schemaLocation in Dataset: No

GML Writer – Import feature types using GML Reader – Application Schema

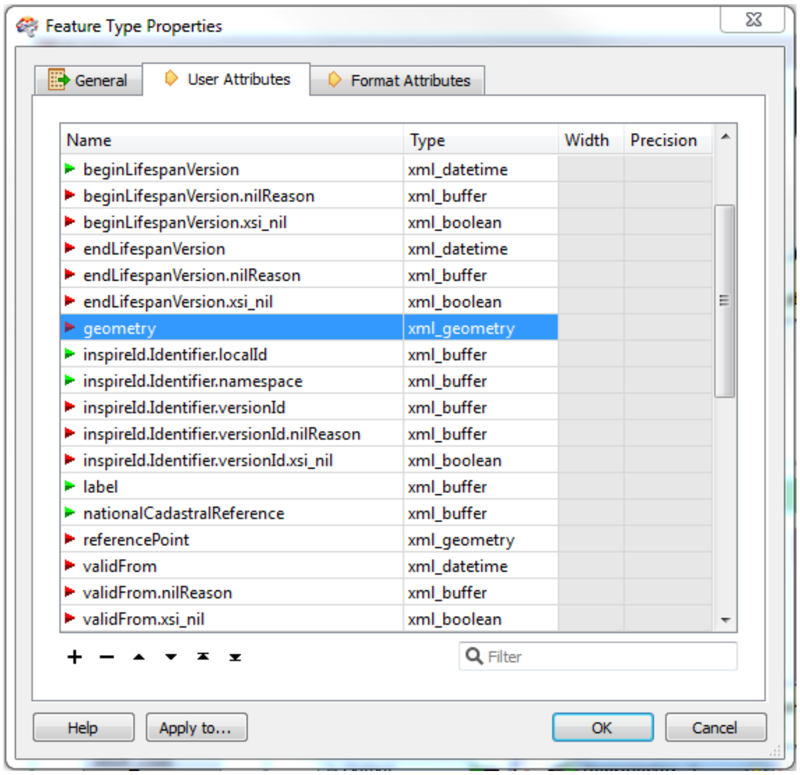
5. When theme browser appears, select only CadastralZoning and CadastralParcel. Connect outputs to the appropriate feature type instead of the inspectors.

6. Set output path.

7. Save and run workspace and inspect output using INSPIRE reader in Data Inspector.

8. Workspace is now similar to INSPIRECadastralParcels-Complete.fmw, and the result similar to output\CadastralParcels.xml

Inspection of destination feature type properties:



Name	Type	Width	Precision
beginLifespanVersion	xml_datetime		
beginLifespanVersion.nilReason	xml_buffer		
beginLifespanVersion.xsi_nil	xml_boolean		
endLifespanVersion	xml_datetime		
endLifespanVersion.nilReason	xml_buffer		
endLifespanVersion.xsi_nil	xml_boolean		
geometry	xml_geometry		
inspireId.Identifier.localId	xml_buffer		
inspireId.Identifier.namespace	xml_buffer		
inspireId.Identifier.versionId	xml_buffer		
inspireId.Identifier.versionId.nilReason	xml_buffer		
inspireId.Identifier.versionId.xsi_nil	xml_boolean		
label	xml_buffer		
nationalCadastralReference	xml_buffer		
referencePoint	xml_geometry		
validFrom	xml_datetime		
validFrom.nilReason	xml_buffer		
validFrom.xsi_nil	xml_boolean		

For more info please see: For more info please see:

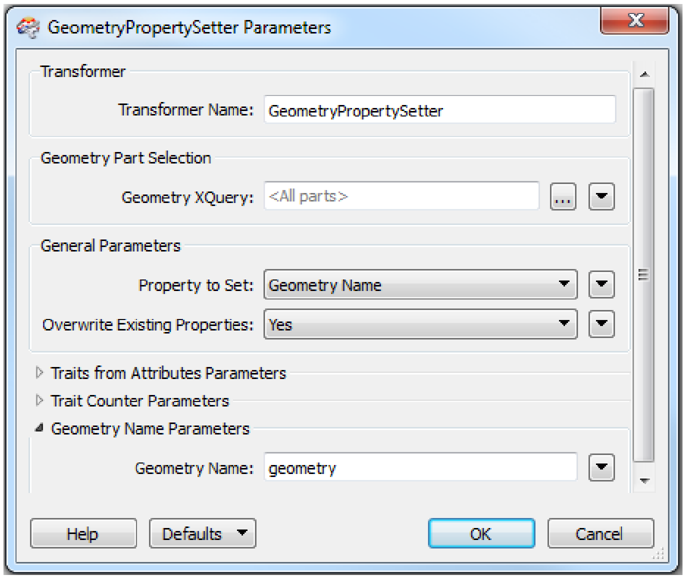
http://fmepedia.safe.com/articles/How_To/GML-Writing

Open the INSPIRE GML Writer parameters, under the User Attribute tabs, note how the complex schema is modeled in terms of:

- parent.child models for nested object relationships, such as inspireId.Identifier.localId
- series{} notation used for series when an element may occur more than once, for example there can be multiple spellings for the same plan name
- Note that gone is the 'text_line_data' field. Now you can do schema mapping to GML the same way as any typical format.
- Key point: note the xml_geometry data type. The output feature's geometry names must correspond to the names of these xml_geometry fields, or else the geometries will be dropped!
- Not all fields must be populated, but some fields are required. Validation will tell you what matters.

Setting Geometry for INSPIRE GML

1. Open the CadastralParcel Writer and under user attributes, look for fields with data type = xml_geometry. Note the field names.
2. Connect the last AttributeCopier output to a GeometryPropertySetter. Open up the GeometryPropertySetter parameters. Select property to set as Geometry Names and set the geometry name to 'geometry'.
3. Connect an InsidePointReplacer after the last AttributeCopier.



GeometryPropertySetter Parameters

Transformer

Transformer Name: GeometryPropertySetter

Geometry Part Selection

Geometry XQuery: <All parts>

General Parameters

Property to Set: Geometry Name

Overwrite Existing Properties: Yes

Traits from Attributes Parameters

Trait Counter Parameters

Geometry Name Parameters

Geometry Name: geometry

4. Connect the InsidePointReplacer output to a second GeometryPropertySetter. Set the geometry name to 'referencePoint'.
5. Connect the outputs from both GeometryPropertySetters to an Aggregator.
6. Set the Aggregator parameters as follows:

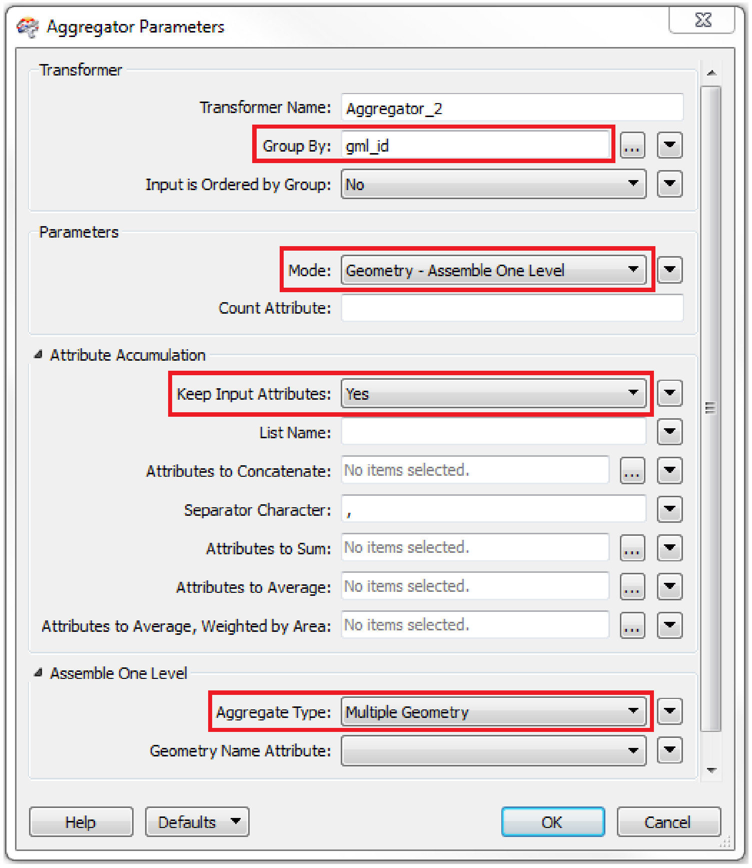
   Group By: gml_id

   Mode: Geometry - Assemble One Level

   Keep Input Attributes: Yes

   Aggregate Type: Multiple Geometry

7. Connect output of Aggregator to CadastralParcel destination feature type.



Aggregator Parameters

Transformer

Transformer Name: Aggregator_2

Group By: gml_id

Input is Ordered by Group: No

Parameters

Mode: Geometry - Assemble One Level

Count Attribute:

Attribute Accumulation

Keep Input Attributes: Yes

List Name:

Attributes to Concatenate: No items selected.

Separator Character: ,

Attributes to Sum: No items selected.

Attributes to Average: No items selected.

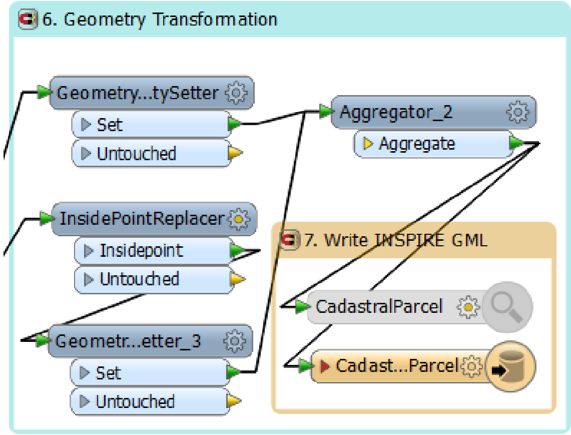
Attributes to Average, Weighted by Area: No items selected.

Assemble One Level

Aggregate Type: Multiple Geometry

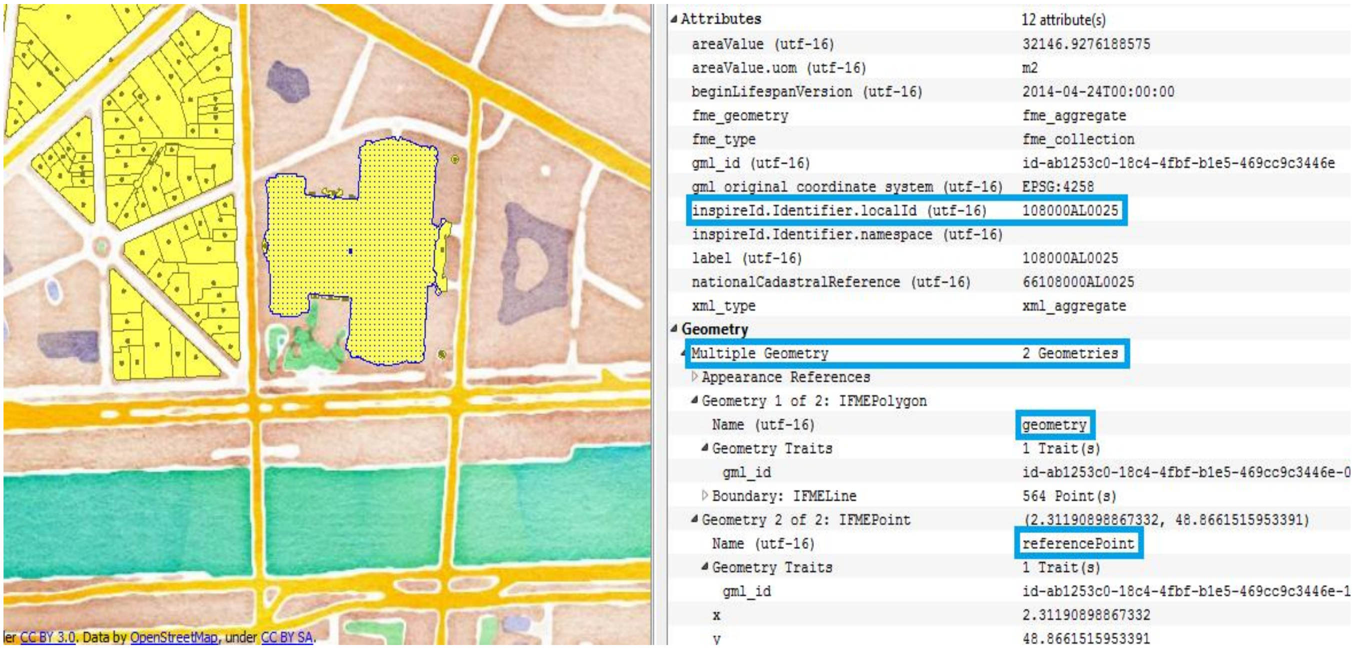
Geometry Name Attribute:

Your full Cadastral Parcels workspace should now look like INSPIRE.CadastralParcels-Complete.fmw. The geometry transformation will look like the image on the right:



Cadastral parcels geometry transformation

8. Rerun the workspace and inspect the output. The output should now be in INSPIRE Cadastral Parcel GML format. Note the nested structure and multiple geometries, 'geometry' and 'referencePoint'.

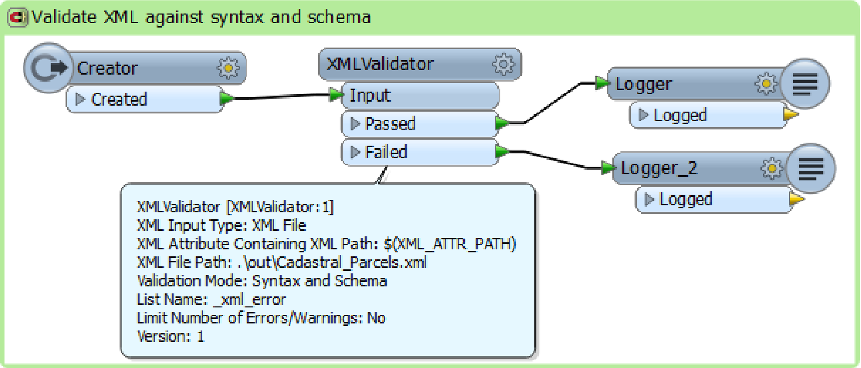


```
◢ Attributes                                       12 attribute(s)
  areaValue (utf-16)                               32146.9276188575
  areaValue.uom (utf-16)                           m2
  beginLifespanVersion (utf-16)                    2014-04-24T00:00:00
  fme_geometry                                     fme_aggregate
  fme_type                                         fme_collection
  gml_id (utf-16)                                  id-ab1253c0-18c4-4fbf-b1e5-469cc9c3446e
  gml_original_coordinate_system (utf-16)          EPSG:4258
  inspireId.Identifier.localId (utf-16)            108000AL0025
  inspireId.Identifier.namespace (utf-16)
  label (utf-16)                                   108000AL0025
  nationalCadastralReference (utf-16)              66108000AL0025
  xml_type                                         xml_aggregate
◢ Geometry
 ◢Multiple Geometry                                2 Geometries
  ▷ Appearance References
  ◢ Geometry 1 of 2: IFMEPolygon
     Name (utf-16)                                 geometry
    ◢ Geometry Traits                              1 Trait(s)
       gml_id                                      id-ab1253c0-18c4-4fbf-b1e5-469cc9c3446e-0
    ▷ Boundary: IFMELine                           564 Point(s)
  ◢ Geometry 2 of 2: IFMEPoint                     (2.31190898867332, 48.8661515953391)
     Name (utf-16)                                 referencePoint
    ◢ Geometry Traits                              1 Trait(s)
       gml_id                                      id-ab1253c0-18c4-4fbf-b1e5-469cc9c3446e-1
     x                                             2.31190898867332
     y                                             48.8661515953391
```

Data Source: Direction générale des Finances Publiques - Cadastre; Updated: 2013

3. INSPIRE Data Validation

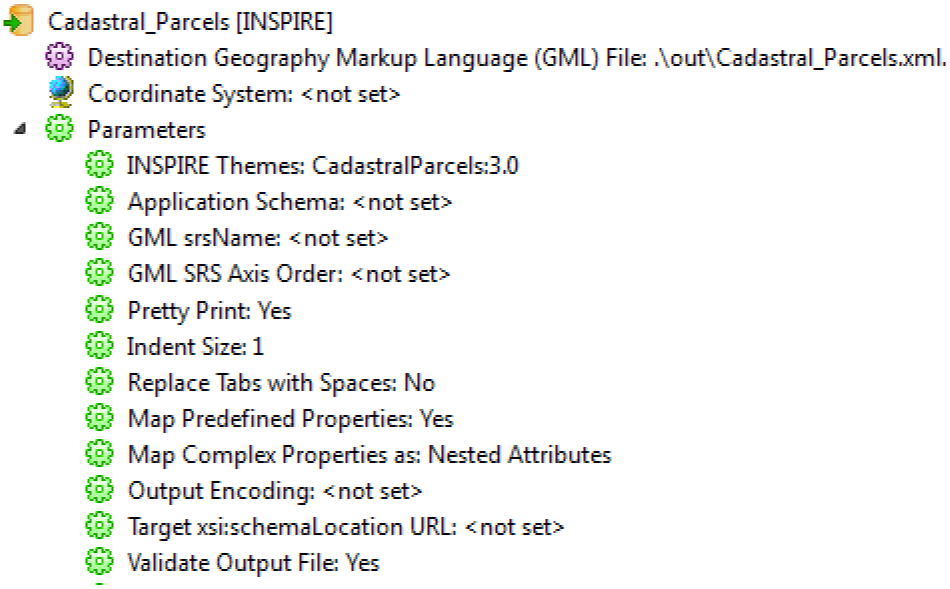
After writing INSPIRE GML, it is important to validate it to ensure that it complies with the XML syntax and XML schema document standards. It is possible to check the GML against syntax and schema using the XMLValidator transformer. If the XML is not valid, in the log, the error or errors will be briefly described, including details such as the line and column number and a brief description of the error. For INSPIRE data, errors are often a result of missing attribute values. Once one has attempted to fix the error, the data can be validated again to check whether the solution worked.



XMLValidator [XMLValidator:1]
XML Input Type: XML File
XML Attribute Containing XML Path: $(XML_ATTR_PATH)
XML File Path: .\out\Cadastral_Parcels.xml
Validation Mode: Syntax and Schema
List Name: _xml_error
Limit Number of Errors/Warnings: No
Version: 1

An example validation workspace

It is also possible to validate the XML output with the INSPIRE Writer. When adding the Writer, the choice to validate output can be selected. Additionally, after creation of an INSPIRE Writer, validation on the output file can be specified in the writer parameters in the Navigator window.



Cadastral_Parcels [INSPIRE]
Destination Geography Markup Language (GML) File: .\out\Cadastral_Parcels.xml.
Coordinate System: <not set>
Parameters
INSPIRE Themes: CadastralParcels:3.0
Application Schema: <not set>
GML srsName: <not set>
GML SRS Axis Order: <not set>
Pretty Print: Yes
Indent Size: 1
Replace Tabs with Spaces: No
Map Predefined Properties: Yes
Map Complex Properties as: Nested Attributes
Output Encoding: <not set>
Target xsi:schemaLocation URL: <not set>
Validate Output File: Yes

Writer parameters in the Navigator window

The image above also exemplifies the ability to turn on Pretty Print, and to select the INSPIRE theme or application schema in the Navigator window. The schema or theme is important for both the writing and validation processes.

Exercise 3:
In this exercise, a validation workspace will be created. The validation workspace will be used to evaluate some good and bad XML.

1. Open a blank workspace.
2. Add a creator, XMLValidator and two Loggers.
3. Open the XMLValidator parameters, and locate CadastralParcels_bad.xml and the CadastralParcels.xsd.
4. Run the workspace and check the log.

Question: The XML fails validation. What is wrong with CadastralParcels_bad.xml?

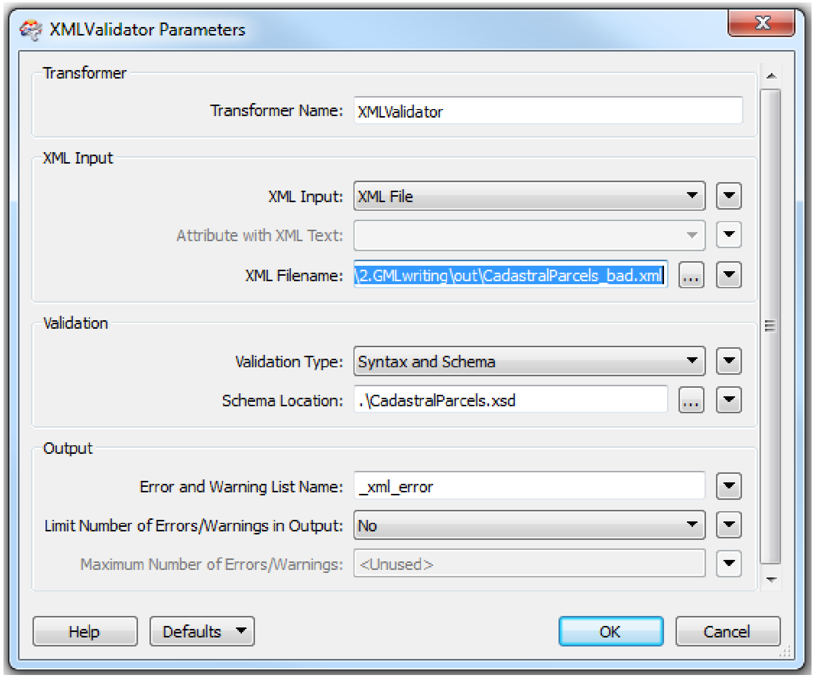
```
`_xml_error{0}.desc' has value `ID value 'id-073c2cd0-1efe-4d4f-aad9-976bf31d6fc6-0' is not unique'
```

Answer: IDs are not unique

5. Now, reopen the XMLValidator parameters, select the XML file CadastralParcels_good.xml.

6. Run the workspace.

7. The XML passes both syntax and schema.



XMLValidator Parameters

Transformer

Transformer Name: XMLValidator

XML Input

XML Input: XML File

Attribute with XML Text:

XML Filename: \2.GMLwriting\out\CadastralParcels_bad.xml

Validation

Validation Type: Syntax and Schema

Schema Location: .\CadastralParcels.xsd

Output

Error and Warning List Name: _xml_error

Limit Number of Errors/Warnings in Output: No

Maximum Number of Errors/Warnings: <Unused>

4. Metadata

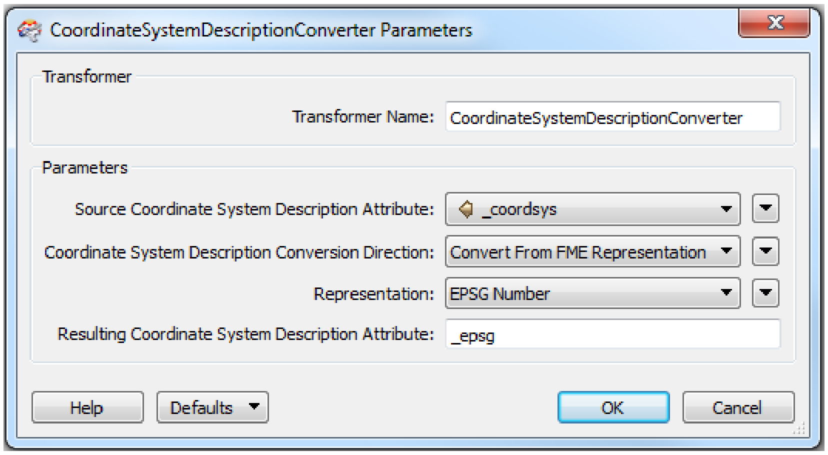
FME support working with metadata, and the format has a large impact on how FME handles it. In the following demo, metadata is automatically harvested from MapInfo TAB files and one ISO19115 XML metadata document is written out for each input file.

Exercise 4:

1. Open the workpace C:\FMEData2014\Resources\INSPIRE\ComplexGML-INSPIREfocus\ReadingGML\Metadata\metadataGeneration-Start.fmw

2. Open the AttributeCreator parameters. Write your name in the myName field. Click ok. The myName attribute is added into the XML by an XMLUpdater, using the XML path:
   `/gmd:MD_Metadata/gmd:contact/gmd:CI_ResponsibleParty/gmd:individualName/gco:CharacterString`

3. After the AttributeCreator transformer, add in a CoordinateSystemExtractor and leave the parameters as default. Add a CoordinateSystemDescriptionConverter, set the source coordinate system description attribute to _coordsys, set representation to EPSG Number and the resulting coordinate system description attribute to _epsg.



CoordinateSystemDescriptionConverter Parameters

Transformer

Transformer Name: CoordinateSystemDescriptionConverter

Parameters

Source Coordinate System Description Attribute: _coordsys

Coordinate System Description Conversion Direction: Convert From FME Representation

Representation: EPSG Number

Resulting Coordinate System Description Attribute: _epsg

CoordinateSystemDescriptionConverter

4. Open XMLTemplater_19115 parameters and expand the template. Near the end of the template expression, locate:

   ```
   <gmd:LI_Lineage>
   <gmd:statement>
   <gco:CharacterString>Purely virtual dataset created from the spirit of the
   Author.
   ```

   Insert in:

   ```
   Dataset uses EPSG:{fme:get-attribute("_epsg")}
   ```

   Before the closing tags:

   ```
   </gco:CharacterString>
   </gmd:statement>
   </gmd:LI_Lineage>
   ```

   The end of the template expression will now look like:

   ```
   <gmd:LI_Lineage>
   <gmd:statement>
   <gco:CharacterString>Purely virtual dataset created from the spirit of the
   Author
   Dataset uses EPSG:{fme:get-attribute("_epsg")}</gco:CharacterString>
   </gmd:statement>
   </gmd:LI_Lineage>
   </gmd:lineage>
   </gmd:DQ_DataQuality>
   </gmd:dataQualityInfo>
   </gmd:MD_Metadata>
   ```

5. For an advanced exercise, try to figure out how to delete the whole facsimile element. Try to follow a similar format to that used to populate the myName attribute, specifying the location of the XML nodes which are to be updated.

View the finished workspace to check your work:

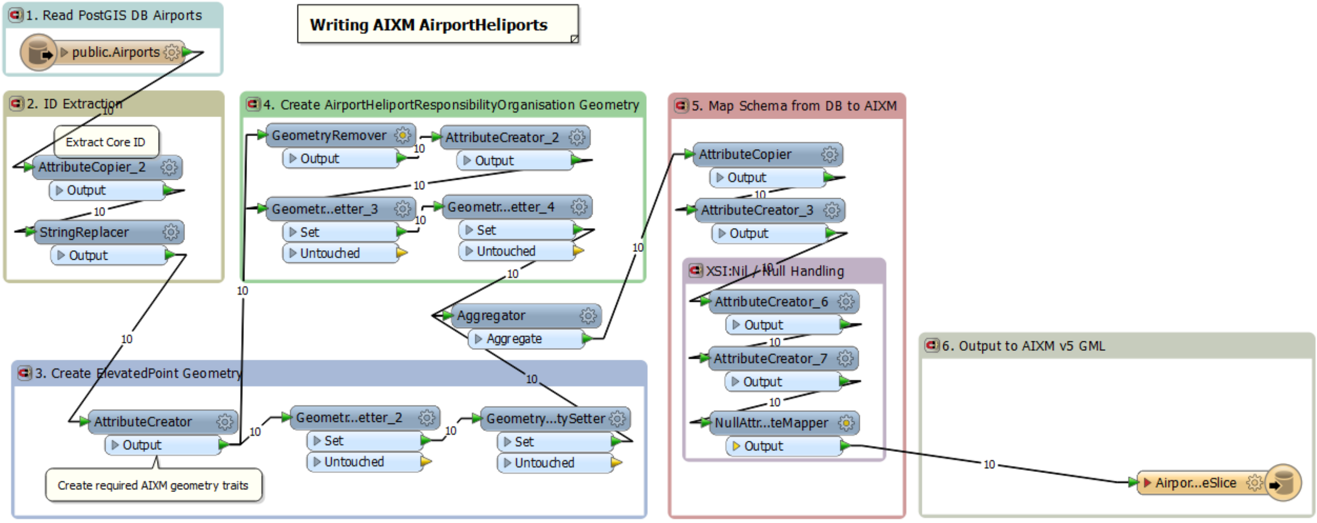
```
C:\FMEData2014\Resources\INSPIRE\ComplexGML-
INSPIREfocus\ReadingGML\Metadata\metadataGeneration-Complete.fmw
```

5. AIXM Airports

Purpose: The goal of this exercise is to read world airports data from an Airports table on PostGIS and transform it so that it can be used to generate AIXM 5 AirportHeliport GML and validate it against the AIXM schemas.

AIXM is one of the most complex GML formats around, even more so than most INSPIRE GML themes. For example, AIXM involves objects within objects, which we model by using feature types that contain other feature types. So correctly defining the relationships between themes is key for this format, as we shall see in this exercise. On top of this, AIXM has a number of unique, complex geometries that have to be correctly defined and annotated in order to successfully create features.

Given the complexity, let's use a starter workspace to begin with as shown below.



airports_postgis2aixm5_Start.fmw

Exercise 5:

In this exercise, data from PostGIS is read in, transformed and written out using the AXIM5 Writer, which results in AirportHeliportTimeSlice AIXM5 data.

1. Open AIXM.Airports\ airports_postgis2aixm5_Start.fmw and run with prompt. Be sure to limit the run to 10 or so features.

2. Open the output in Data Inspector and in an XML editor.

3. Turn on validation on the AIXM writer and rerun. What problems do you encounter?

AIXM Diagnostics

```
ERROR |XML Validation: Error in
'\\bester\pserv\Projects\INSPIRE\Training\FMEUC2014_training\ComplexGML-
INSPIREfocus\AIXM.Airports\output\AerodromeNodes.xml' on line 313, column 46: 'value 'El Gran
Roque / Ntl' does not match regular expression facet '([A-Z]|[0-9]|[, !"&#$%'\(\)\*\+\-
\./:;<=>\?@\[\\\]\^_\|\{\}])*''
ERROR |XML Validation: Error in
'\\bester\pserv\Projects\INSPIRE\Training\FMEUC2014_training\ComplexGML-
INSPIREfocus\AIXM.Airports\output\AerodromeNodes.xml' on line 327, column 35: 'element
'sequenceNumber' is not allowed for content model
'(validTime,interpretation,sequenceNumber?,correctionNumber?,timeSliceMetadata?,featureLifetime?,
designator?,name?,locationIndicatorICAO?,designatorIATA?,type?,certifiedICAO?,privateUse?,control
Type?,fieldElevation?,fieldElevationAccuracy?,verticalDatum?,magneticVariation?,magneticVariation
Accuracy?,dateMagneticVariation?,magneticVariationChange?,referenceTemperature?,altimeterCheckLoc
ation?,secondaryPowerSupply?,windDirectionIndicator?,landingDirectionIndicator?,transitionAltitud
e?,transitionLevel?,lowestTemperature?,abandoned?,certificationDate?,certificationExpirationDate?
,contaminant*,servedCity*,responsibleOrganisation?,ARP?,aviationBoundary?,altimeterSource*,contac
t*,availability*,annotation*,extension*)''
ERROR |XML Validation: Error in
'\\bester\pserv\Projects\INSPIRE\Training\FMEUC2014_training\ComplexGML-
INSPIREfocus\AIXM.Airports\output\AerodromeNodes.xml' on line 328, column 22: 'element
'AirportHeliportTimeSlice' is not allowed for content model '(AbstractFeature)''
```

Validation error log from running airports_postgis2aixm5_Start.fmw

Given all the red ERRORs, this may look rather scary. It's actually looks worse than it is, especially once you get used to it. Let's decode this before we go any further.

```
'value 'El Gran Roque / Ntl' does not match regular expression facet '([A-Z]|[0-9]|[,
!"&#$%'\(\)\*\+\-\./:;<=>\?@\[\\\]\^_\|\{\}])*''
```

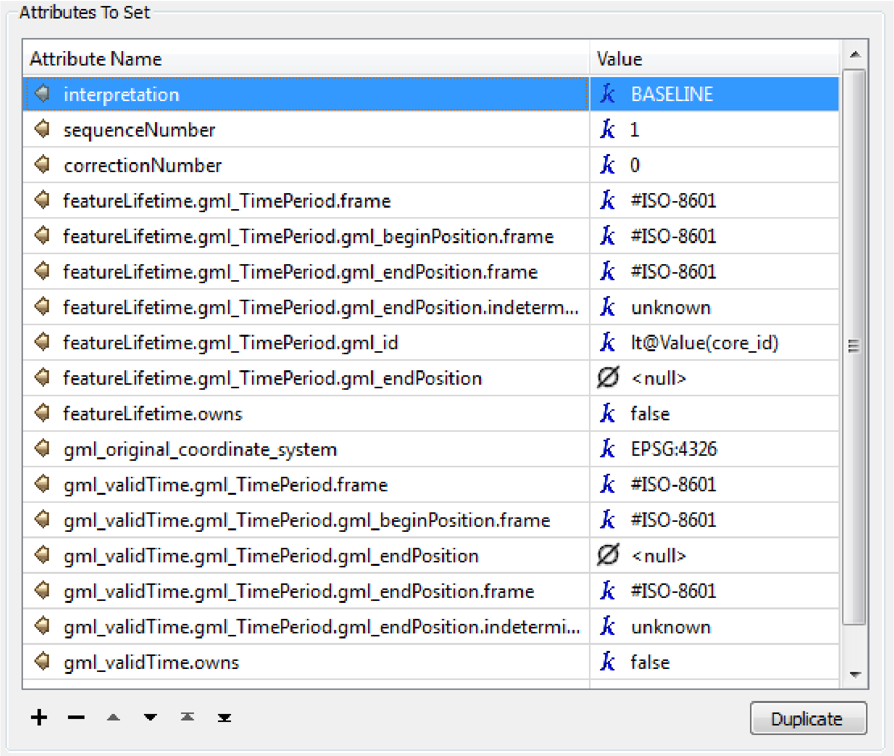
This simply means that the schema does not allow for mixed case ([A-Z] only), so we will need to force to upper case for the airport name field.

```
'sequenceNumber' is not allowed for content model
```

This is a bit trickier. While these messages that we get back from the Xerxes parser are sometimes a bit obscure, they are consistent. Typically these 'not allowed for content model' messages appear when something is missing, rather than something extra that shouldn't be there which the message suggests. To find out what is missing we need to go through the field list and look for things that are not ? or *. Deleting all of the wildcarded fields, we are left with 'validTime and interpretation'. Everything else is optional. By examining our output we can see that only interpretation is missing. So we will need to add this field.

```
'element 'AirportHeliportTimeSlice' is not allowed for content model '(AbstractFeature)''
```

This is similar to the above error except that in this case an entire element or object is missing. So it isn't so much that 'AirportHeliportTimeSlice' is not allowed, its just that it isn't allowed on its own – it must belong to a parent AirportHeliport. So we will need to add this new output feature type and populate it with a parent feature if we want to avoid this error.



Attributes To Set

Attribute Name	Value
interpretation	BASELINE
sequenceNumber	1
correctionNumber	0
featureLifetime.gml_TimePeriod.frame	#ISO-8601
featureLifetime.gml_TimePeriod.gml_beginPosition.frame	#ISO-8601
featureLifetime.gml_TimePeriod.gml_endPosition.frame	#ISO-8601
featureLifetime.gml_TimePeriod.gml_endPosition.indeterm...	unknown
featureLifetime.gml_TimePeriod.gml_id	lt@Value(core_id)
featureLifetime.gml_TimePeriod.gml_endPosition	<null>
featureLifetime.owns	false
gml_original_coordinate_system	EPSG:4326
gml_validTime.gml_TimePeriod.frame	#ISO-8601
gml_validTime.gml_TimePeriod.gml_beginPosition.frame	#ISO-8601
gml_validTime.gml_TimePeriod.gml_endPosition	<null>
gml_validTime.gml_TimePeriod.gml_endPosition.frame	#ISO-8601
gml_validTime.gml_TimePeriod.gml_endPosition.indetermi...	unknown
gml_validTime.owns	false

AttributeCreator_3 AIXM field definitions

1. Open AttributeCreator_3 and add a definition for: Interpretation = BASELINE

2. Add a StringCaseChanger after the NullAttributeMapper to correct the mixed case problem. Select the name attribute and set the case to UPPERCASE.

3. Rerun the workspace to confirm which error this fixes.

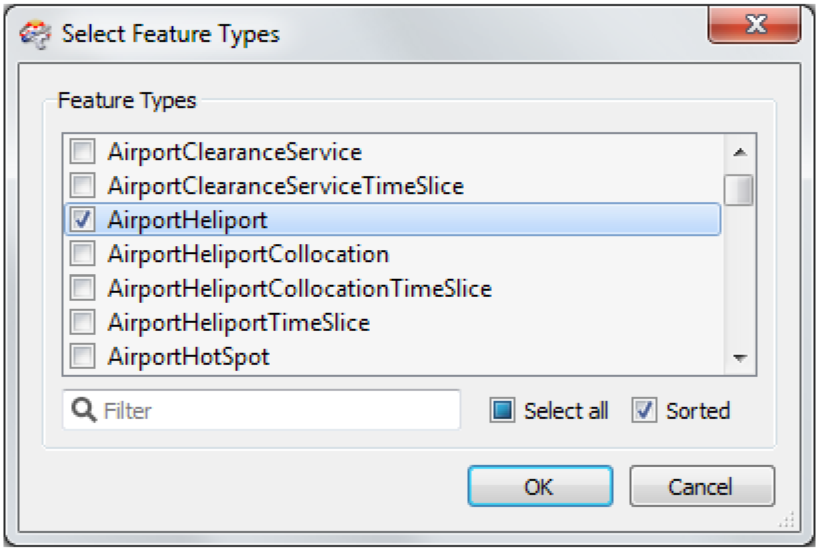
You should now only have the 'AirportHeliportTimeSlice' is not allowed error. To fix this will take a little more work, since we need to create a parent object for AirportHeliportTimeSlice. To do this, do the following:

1. After the StringCaseChanger add an AttributeCreator and define: gml_parent_id = parent@Value(gml_id). Make sure the output of this transformer still connects to the AirportHeliportTimeSlice output feature type.

2. Also after the StringCaseChanger add another AttributeCreator (parallel but not connected to the one above) and define: gml _id = parent@Value(gml_id). This essentially makes one copy of each airport point so we can create one parent feature for each point.

3. Go to the Writer menu and select Import Feature Type. Choose AIXM5 reader and in the reader settings choose Application Schema AIXM Version: http://www.aixm.aero/schema/5.1/message

   A list of feature types will appear. Choose only AirportHeliport.



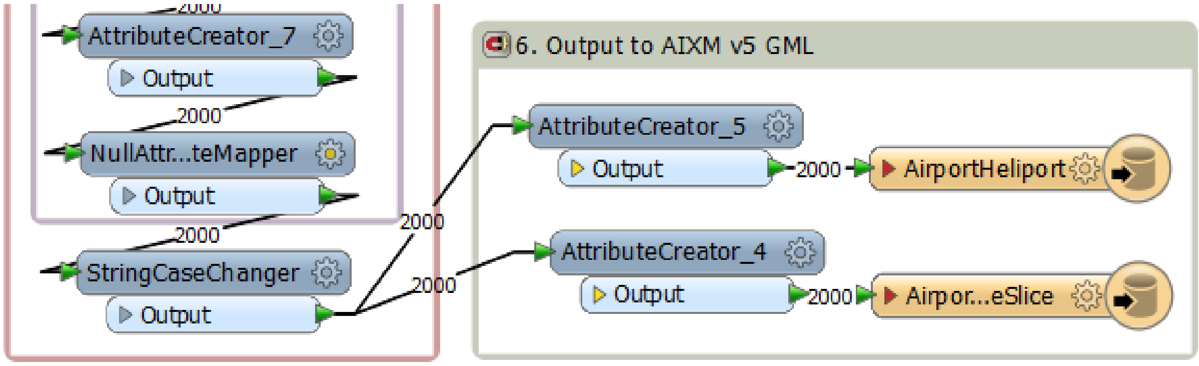
Select Feature Types

Feature Types

☐ AirportClearanceService
☐ AirportClearanceServiceTimeSlice
☑ AirportHeliport
☐ AirportHeliportCollocation
☐ AirportHeliportCollocationTimeSlice
☐ AirportHeliportTimeSlice
☐ AirportHotSpot

Import AIXM feature types

4. Connect the last AttributeCreator you just added (AttributeCreator_5) with gml _id = parent@Value(gml_id) to the AirportHeliport output feature type. The output section of your completed workspace should look something like the image below.



Output section of airports_postgis2aixm5_Completed.fmw

5. Rerun the workspace. Do you still see any red ERROR messages?

This workflow is not contrived but rather mimics closely the process we typically go through when trying to write to new application schemas that we have not encountered before. The trick is to limit the number of features, work at a few feature types at a time, have documentation available to explain schema requirements, and learn how to read the Xerxes error messages. We plan on publishing new articles about XML diagnostics soon to FMEpedia, so stay tuned for more info on this subject.

6. Converting from CityGML to INSPIRE Buildings

Purpose: Convert 3D buildings from CityGML to INSPIRE Annex III Buildings

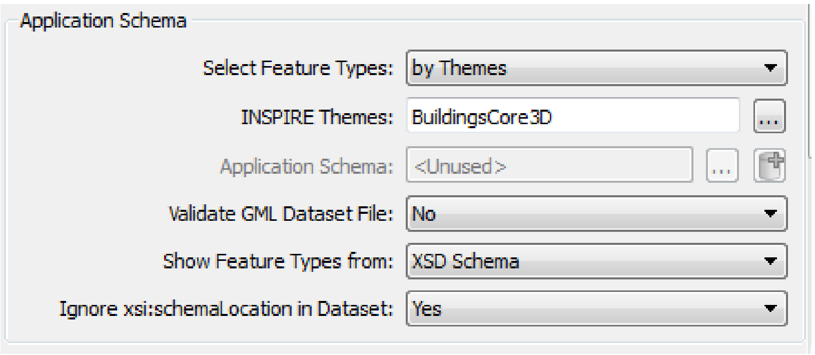
Workflow Overview: Start with a default translation, then add schema mapping and geometry restructuring.

1. Open a new workspace and add CityGML Reader:

   Reader Format: CityGML
   Reader Dataset: …\source\Buildings\10-17-KOP_VAN_ZUID.xml
   Select source feature types: GroundSurface, WallSurface, RoofSurface

2. Add INSPIRE GML Writer:

   Writer Format: INSPIRE GML
   Writer Dataset: .\output\INSPIREbuildings.xml
   Select destination feature type: BuildingsCore3D
   Say No to 'add New Feature Type'
   Go to **Writer – Import Feature Types – INSPIRE GML** - Reader Parameters and in the Application Schema section set Select Feature Types: by Themes, and INSPIRE Themes: to BuildingsCore3D

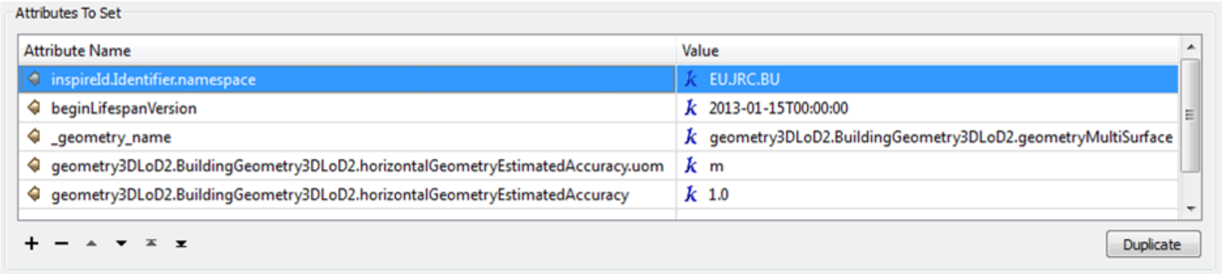


Application Schema

Select Feature Types: by Themes
INSPIRE Themes: BuildingsCore3D
Application Schema: <Unused>
Validate GML Dataset File: No
Show Feature Types from: XSD Schema
Ignore xsi:schemaLocation in Dataset: Yes

INSPIRE GML Reader Application Schema settings

3. Add an **AttributeCreator**. Connect all source feature types to it. Create the required INSPIRE buildings fields as follows:



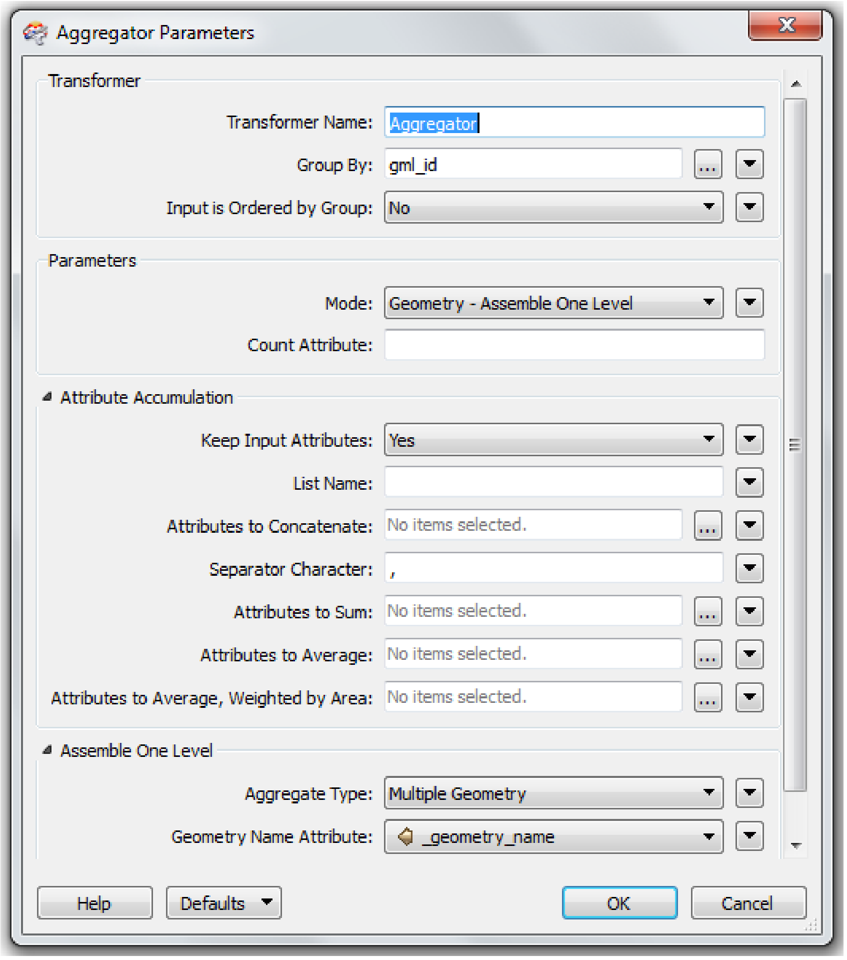
Attributes To Set

Attribute Name	Value
inspireId.Identifier.namespace	EUJRC.BU
beginLifespanVersion	2013-01-15T00:00:00
_geometry_name	geometry3DLoD2.BuildingGeometry3DLoD2.geometryMultiSurface
geometry3DLoD2.BuildingGeometry3DLoD2.horizontalGeometryEstimatedAccuracy.uom	m
geometry3DLoD2.BuildingGeometry3DLoD2.horizontalGeometryEstimatedAccuracy	1.0

AttributeCreator settings to define required INSPIRE buildings fields.

4. Add an **AttributeCopier** to remap source field gml_id to destination field inspireId.Identifier.localId

5. Add a **GeometryPropertyRemover** with the default settings. Property to remove: Traits; Remove all: Yes
6. Add an **Aggregator**. Configure it as follows:



Aggregator Parameters

Transformer

Transformer Name: Aggregator

Group By: gml_id

Input is Ordered by Group: No

Parameters

Mode: Geometry - Assemble One Level

Count Attribute:

Attribute Accumulation

Keep Input Attributes: Yes

List Name:

Attributes to Concatenate: No items selected.

Separator Character: ,

Attributes to Sum: No items selected.

Attributes to Average: No items selected.

Attributes to Average, Weighted by Area: No items selected.

Assemble One Level

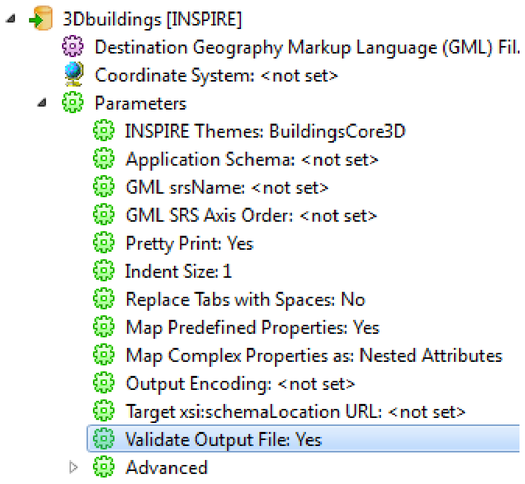
Aggregate Type: Multiple Geometry

Geometry Name Attribute: _geometry_name

Connect the output of the Aggregator to the BuildingPart output feature type.

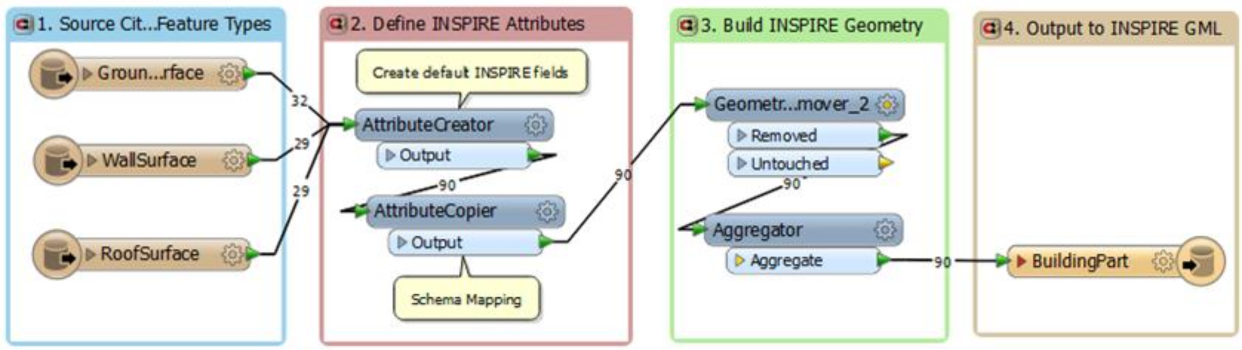
7. Set the INSPIRE writer settings as follows opposite.

   Note the INSPIRE theme, pretty print and validate output settings.

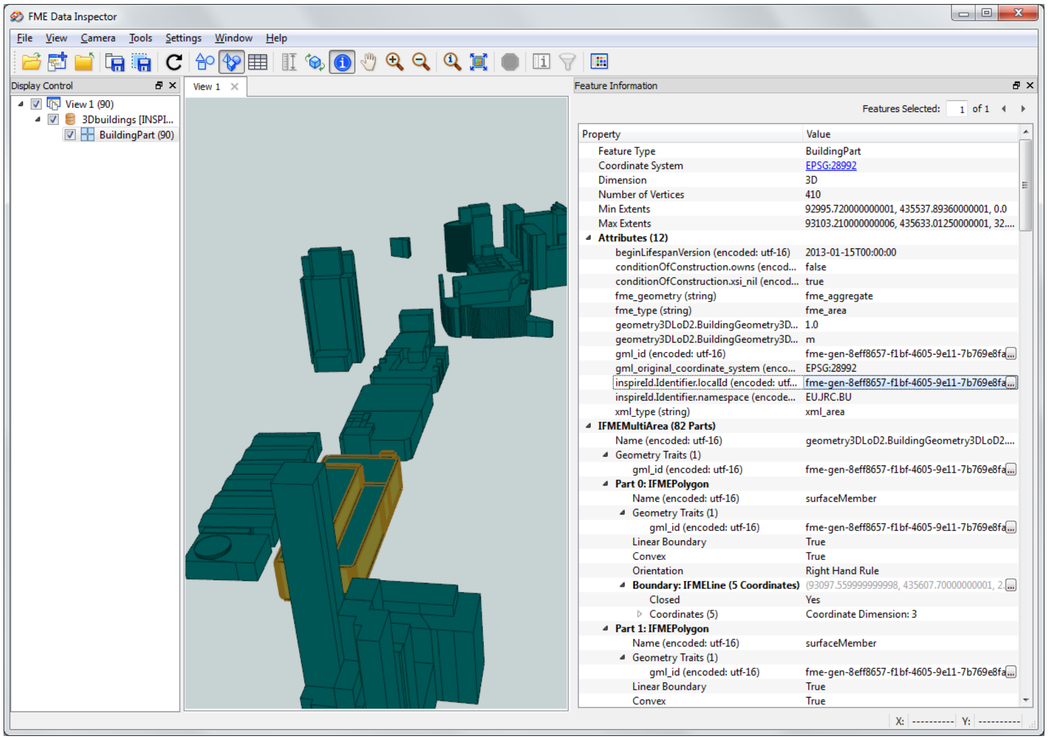


- 3Dbuildings [INSPIRE]
  - Destination Geography Markup Language (GML) Fil.
  - Coordinate System: <not set>
  - Parameters
    - INSPIRE Themes: BuildingsCore3D
    - Application Schema: <not set>
    - GML srsName: <not set>
    - GML SRS Axis Order: <not set>
    - Pretty Print: Yes
    - Indent Size: 1
    - Replace Tabs with Spaces: No
    - Map Predefined Properties: Yes
    - Map Complex Properties as: Nested Attributes
    - Output Encoding: <not set>
    - Target xsi:schemaLocation URL: <not set>
    - Validate Output File: Yes
    - Advanced

8. Run the translation. What happens? What warnings do you see in the log? Are you able to read the output data with Data Inspector?



Completed CityGML to INSPIRE Buildings Workspace



Features Selected: 1 of 1

Property	Value
Feature Type	BuildingPart
Coordinate System	EPSG:28992
Dimension	3D
Number of Vertices	410
Min Extents	92995.720000000001, 435537.89360000001, 0.0
Max Extents	93103.210000000006, 435633.01250000001, 32....
Attributes (12)	
beginLifespanVersion (encoded: utf-16)	2013-01-15T00:00:00
conditionOfConstruction.owns (encod...	false
conditionOfConstruction.xsi_nil (encod...	true
fme_geometry (string)	fme_aggregate
fme_type (string)	fme_area
geometry3DLoD2.BuildingGeometry3D...	1.0
geometry3DLoD2.BuildingGeometry3D...	m
gml_id (encoded: utf-16)	fme-gen-8eff8657-f1bf-4605-9e11-7b769e8fa...
gml_original_coordinate_system (enco...	EPSG:28992
inspireId.Identifier.localId (encoded: utf...	fme-gen-8eff8657-f1bf-4605-9e11-7b769e8fa...
inspireId.Identifier.namespace (encode...	EU.JRC.BU
xml_type (string)	xml_area
IFMEMultiArea (82 Parts)	
Name (encoded: utf-16)	geometry3DLoD2.BuildingGeometry3DLoD2....
Geometry Traits (1)	
gml_id (encoded: utf-16)	fme-gen-8eff8657-f1bf-4605-9e11-7b769e8fa...
Part 0: IFMEPolygon	
Name (encoded: utf-16)	surfaceMember
Geometry Traits (1)	
gml_id (encoded: utf-16)	fme-gen-8eff8657-f1bf-4605-9e11-7b769e8fa...
Linear Boundary	True
Convex	True
Orientation	Right Hand Rule
Boundary: IFMELine (5 Coordinates)	(93097.559999999998, 435607.70000000001, 2...
Closed	Yes
Coordinates (5)	Coordinate Dimension: 3
Part 1: IFMEPolygon	
Name (encoded: utf-16)	surfaceMember
Geometry Traits (1)	
gml_id (encoded: utf-16)	fme-gen-8eff8657-f1bf-4605-9e11-7b769e8fa...
Linear Boundary	True
Convex	True

INSPIRE Buildings GML as viewed in Data Inspector

Bonus Exercise

You can see in the log the following warning:

```
'90' orphaned features found, these features will be appended as children to
the root element
```

What is the problem here and how do you think we can address it?

In general we should make sure we are writing the parent objects that these building parts should belong to. To do this, Import the source feature types: Building and CityModel and the destination feature types: Building and SpatialDataset. Try to figure out the attribute mappings and new fields required to generate adequate parent objects for the building parts so this warning no longer appears. You may need to refer to the INSPIRE building specifications to better understand the required relationships (parent / child ids etc).

Hints:

- We don't need to duplicate the geometries here so you may want to start with a GeometryRemover.
- You will need to decide whether building parts are children of Buildings or SpatialDataSet. Once you decide this you will need to modify their gml_parent_ids to match the parent object(s) gml_id(s).

7. INSPIRE List Elements

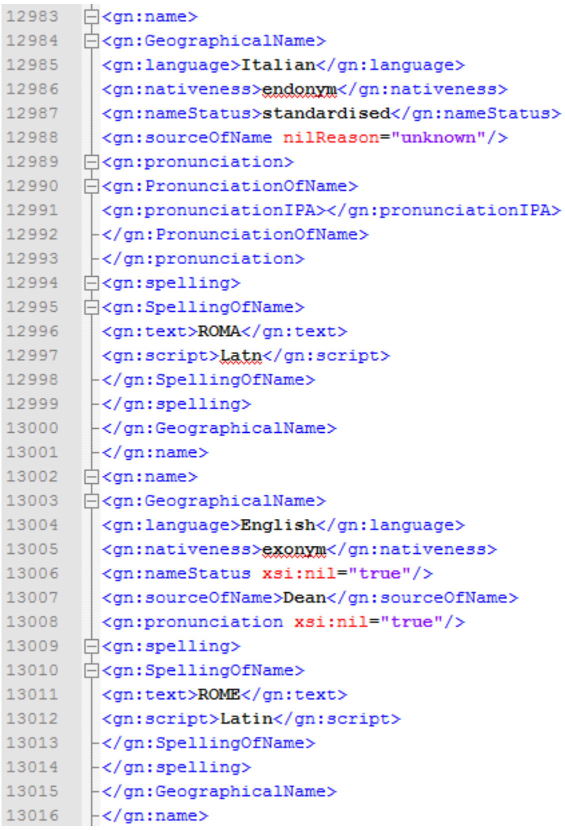
In the INSPIRE schemas, the list{} notation is used for series when an element can occur more than once.

Exercise 7:

This demo analyzes INSPIRE compliant Geographic Names GML 3.2.1 data using a XML or text editor, as well as Data Inspector.

1. Download Geographical Names demo from FMEpedia.
2. Run it.
3. Open the output in an XML or text editor and find the one point that has 2 spellings – one in English and one in Italian.

Question: Where is the point with 2 spellings?

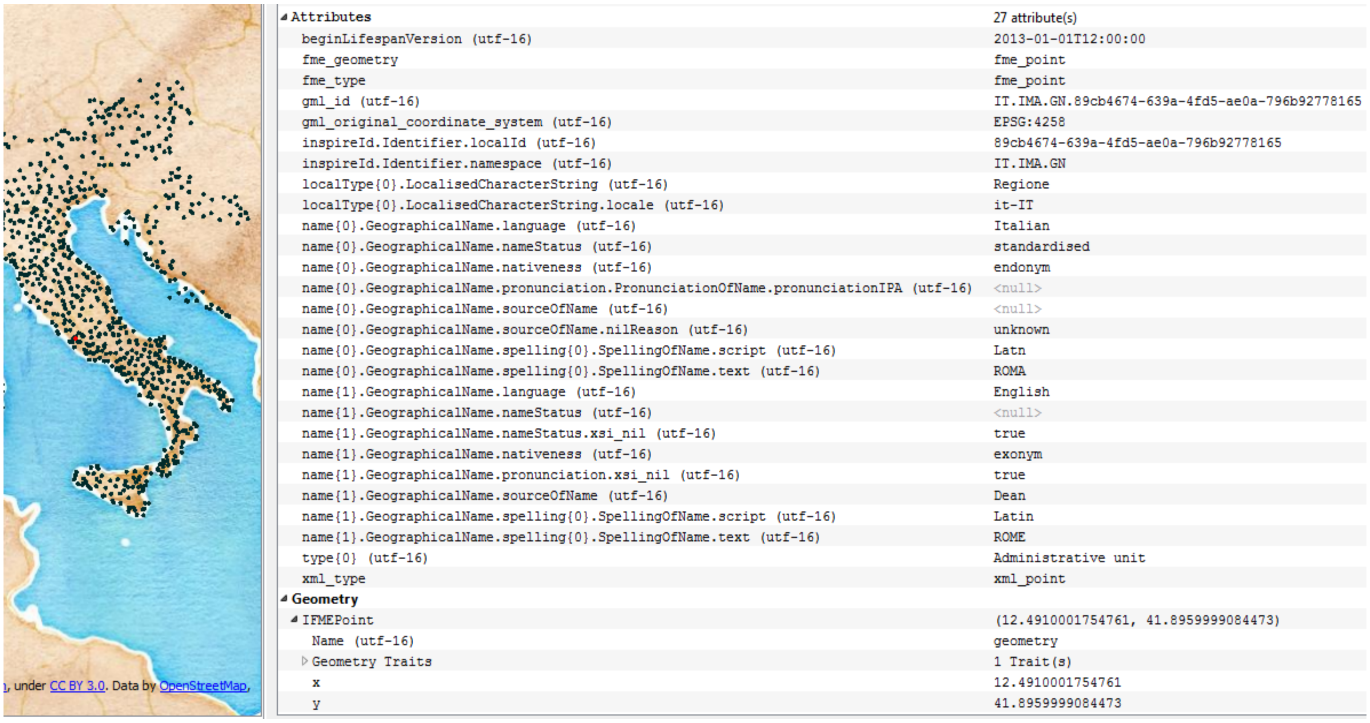


```
<gn:name>
<gn:GeographicalName>
<gn:language>Italian</gn:language>
<gn:nativeness>endonym</gn:nativeness>
<gn:nameStatus>standardised</gn:nameStatus>
<gn:sourceOfName nilReason="unknown"/>
<gn:pronunciation>
<gn:PronunciationOfName>
<gn:pronunciationIPA></gn:pronunciationIPA>
</gn:PronunciationOfName>
</gn:pronunciation>
<gn:spelling>
<gn:SpellingOfName>
<gn:text>ROMA</gn:text>
<gn:script>Latn</gn:script>
</gn:SpellingOfName>
</gn:spelling>
</gn:GeographicalName>
</gn:name>
<gn:name>
<gn:GeographicalName>
<gn:language>English</gn:language>
<gn:nativeness>exonym</gn:nativeness>
<gn:nameStatus xsi:nil="true"/>
<gn:sourceOfName>Dean</gn:sourceOfName>
<gn:pronunciation xsi:nil="true"/>
<gn:spelling>
<gn:SpellingOfName>
<gn:text>ROME</gn:text>
<gn:script>Latin</gn:script>
</gn:SpellingOfName>
</gn:spelling>
</gn:GeographicalName>
</gn:name>
```

Answer: Point with SpellingOfName.text of Rome and Roma

4. Open the output with Inspector. Find and select the city.

Question: How are the multiple place names stored on the same feature?



```
◢ Attributes                                                                       27 attribute(s)
   beginLifespanVersion (utf-16)                                                   2013-01-01T12:00:00
   fme_geometry                                                                    fme_point
   fme_type                                                                        fme_point
   gml_id (utf-16)                                                                 IT.IMA.GN.89cb4674-639a-4fd5-ae0a-796b92778165
   gml_original_coordinate_system (utf-16)                                         EPSG:4258
   inspireId.Identifier.localId (utf-16)                                           89cb4674-639a-4fd5-ae0a-796b92778165
   inspireId.Identifier.namespace (utf-16)                                         IT.IMA.GN
   localType{0}.LocalisedCharacterString (utf-16)                                  Regione
   localType{0}.LocalisedCharacterString.locale (utf-16)                           it-IT
   name{0}.GeographicalName.language (utf-16)                                      Italian
   name{0}.GeographicalName.nameStatus (utf-16)                                    standardised
   name{0}.GeographicalName.nativeness (utf-16)                                    endonym
   name{0}.GeographicalName.pronunciation.PronunciationOfName.pronunciationIPA (utf-16) <null>
   name{0}.GeographicalName.sourceOfName (utf-16)                                  <null>
   name{0}.GeographicalName.sourceOfName.nilReason (utf-16)                        unknown
   name{0}.GeographicalName.spelling{0}.SpellingOfName.script (utf-16)             Latn
   name{0}.GeographicalName.spelling{0}.SpellingOfName.text (utf-16)               ROMA
   name{1}.GeographicalName.language (utf-16)                                      English
   name{1}.GeographicalName.nameStatus (utf-16)                                    <null>
   name{1}.GeographicalName.nameStatus.xsi_nil (utf-16)                            true
   name{1}.GeographicalName.nativeness (utf-16)                                    exonym
   name{1}.GeographicalName.pronunciation.xsi_nil (utf-16)                         true
   name{1}.GeographicalName.sourceOfName (utf-16)                                  Dean
   name{1}.GeographicalName.spelling{0}.SpellingOfName.script (utf-16)             Latin
   name{1}.GeographicalName.spelling{0}.SpellingOfName.text (utf-16)               ROME
   type{0} (utf-16)                                                                Administrative unit
   xml_type                                                                        xml_point
◢ Geometry
 ◢ IFMEPoint                                                                       (12.4910001754761, 41.8959999084473)
    Name (utf-16)                                                                  geometry
  ▷ Geometry Traits                                                                1 Trait(s)
    x                                                                              12.4910001754761
    y                                                                              41.8959999084473
```

List structure for Rome in INSPIRE Geographical Names

Answer: As multiple elements in a list attribute, meaning that each name spelling is a separate element.

Further Resources

If you are interested in learning more about using FME to work with INSPIRE data, please check out the following FMEpedia articles. Also, more demos are in progress, so search INSPIRE on FMEpedia for the latest.

The Cadastral Parcels Writing demo is a good place to start. It covers the typical workflow to write INSPIRE compliant GML, which can be applied to other INSPIRE themes. The workspace reads French national cadastre data and writes out INSPIRE GML according to the INSPIRE Annex I Cadastral Parcels schema (XSD). It also covers geometry transformation from simple single-part geometry to complex multi-part geometry.

The INSPIRE Geographic Names demo generates INSPIRE compliant Geographic Names GML 3.2.1 data from UN gazetteer shapefile data. This example includes combining multiple data sources, as well as two different methods of schema mapping.

There will also be many more new articles added over the course of the coming year, so check back at http://fmepedia.safe.com/topic/XML/GML/KML/Web and http://fmepedia.safe.com/articles/How_To/INSPIRE-GML to see what's new.

SAFE SOFTWARE™

Safe Software values its customers' opinions very highly.
For questions and concerns, please use the general feedback page at: **www.safe.com/contact**, or
email the Training Manager directly at: **training@safe.com**.