

## 4. INTERLIS-Anwendertreffen

# **Workshop INTERLIS, FME und ESRI**

## Übungen



Chemin de Maillefer 36  
1052 Le Mont-sur-Lausanne  
Tel. 021 643 77 11  
[info@inser.ch](mailto:info@inser.ch)

Autor : AV  
Version : 1.0 vom 05.11.2024

## INHALTSVERZEICHNIS

<b>1.</b>	<b>BSP. 1 - EINLESEN EINER INTERLIS-DATEI 2 .....</b>	<b>3</b>
1.1	Überblick .....	3
1.2	Erstellen des FME-Workspace .....	3
<b>2.</b>	<b>Ü 2 - DATEN SCHREIBEN INTERLIS 2 .....</b>	<b>7</b>
2.1	Einführung .....	7
2.2	Daten .....	7
2.3	Entwicklung des Workspace .....	7
2.3.1	Definition des INTERLIS 2 Writers: .....	8
<b>3.</b>	<b>Ü 3 - MGDM LESEN UND EXPORT EN GDB.....</b>	<b>12</b>
3.1	Einführung .....	12
3.2	Ziele .....	12
3.3	Daten des Geschäftsjahres .....	12
3.4	Vorgehensweise .....	12
<b>4.</b>	<b>Ü 4 - OPTIONAL - MGDM-EXPORT MIT FME.....</b>	<b>18</b>
4.1	Einführung .....	18
4.2	Ziele .....	18
4.3	Vorgehensweise .....	18

## 1. ÜBUNG 1 - EINLESEN EINER INTERLIS-2-DATEI

### 1.1 Präsentation

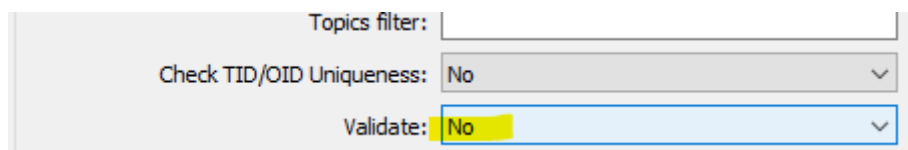
Für diese Übung steht uns eine INTERLIS 2.3-Datei zur Verfügung, die Daten zu den Kantonen enthält (beschreibende sowie geometrische Daten). Ein Kanton kann aus mehreren Teilen bestehen (Hauptform und Enklave). Die Geometrie ist daher mehrteilig.

Ziel ist es, die Interlis-Datei zu lesen und die Geometrie in der entsprechenden Struktur zu suchen.

### 1.2 Erstellen des FME-Workspace

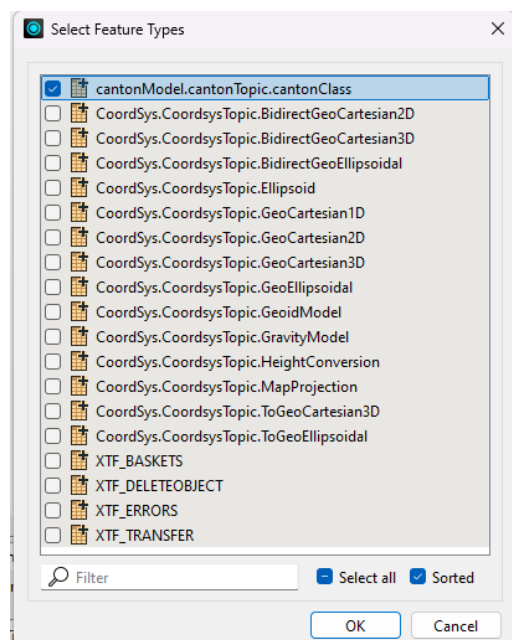
Interlis 2.3-Datei einlesen :

1. Auf der Registerkarte Readers wählen Sie Add reader aus
2. Das Format wird *Swiss INTERLIS (ili2fme)* sein und es muss die Datei *cantons.xtf* als Data Set ausgewählt werden.
3. Wenn Sie auf die Schaltfläche *Parameters* klicken, müssen Sie die Funktion *Validate* deaktivieren. Für die Übung spart das Zeit beim Einlesen der xtf-Datei.

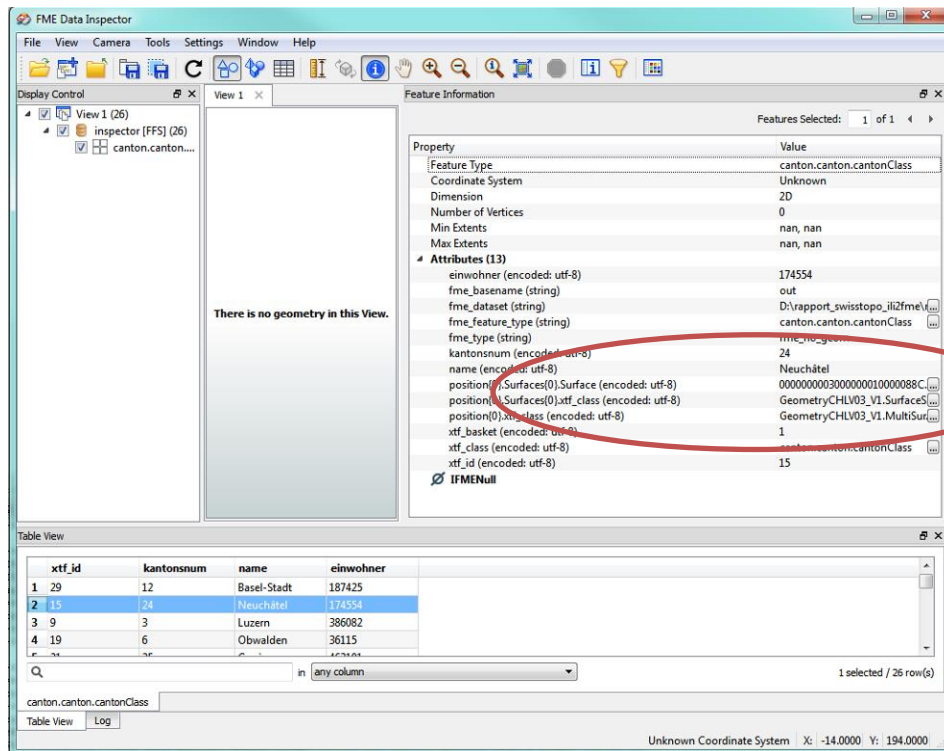


Bestätigen Sie beide Fenster mit OK, um die Daten zu importieren.

4. FME bietet den Import aller Klassen an, die in unseren Datenmodellen vorhanden sind (das allgemeine, *cantonModel.ili*, sowie alle, die über das *IMPORT-Tag* importiert wurden). In diesem Fall werden wir nur die Klasse: *cantonClass* importieren.



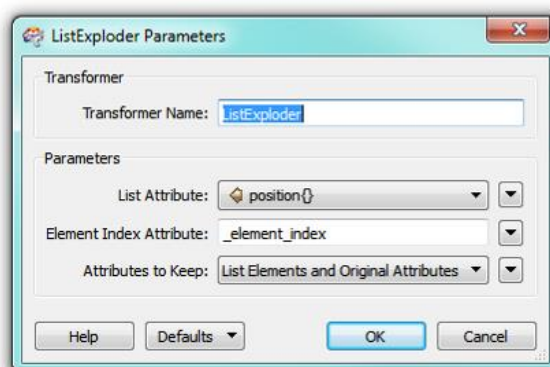
5. Wenn wir einen Inspektor mit dem importierten Reader verbinden, erhalten wir keine Geometrie. Wir können jedoch feststellen, dass vier Attribute vorhanden sind. Das erste Attribut entspricht der *xtf\_id* und die folgenden sind die beschreibenden Attribute der Kantone. Die Geometrie wird in Form einer Liste gespeichert. Auf diese Weise ermöglicht ili2fme die Gruppierung von Multipart-Geometrien. Wenn wir auf einen Wert der angezeigten Attribute klicken, können wir sie im Fenster Feature Information sehen.



Um diese Informationen nutzen zu können, müssen diese Listen also extrahiert werden.

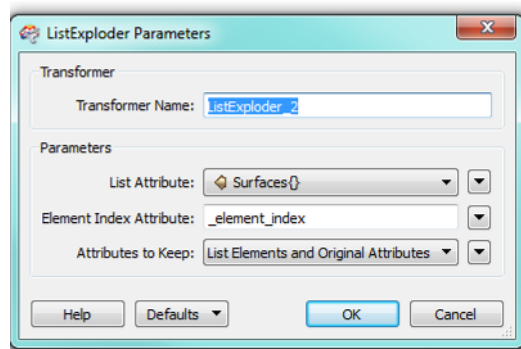
Beachten Sie, dass im Falle einer XTF, bei der die Geometrie nicht multi-part ist, diese direkt vom ili2fme-Plugin erkannt wird.

6. Dann werden wir mithilfe eines ListExploders die Liste position{ } exponieren, was dem Namen des Geometrieattributs im ili-Modell entspricht.

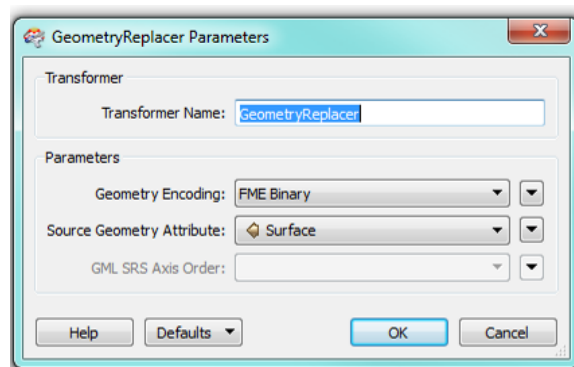


Wenn wir einen Inspektor wieder mit der Ausgabe dieses ListExploders verbinden, können wir sehen, dass das Attribut xtf\_class erstellt wurde. Die Werte dieses Attributs entsprechen der Geometrieklasse

der Kantone. Allerdings ist es immer noch nicht möglich, die Geometrie zu visualisieren. Dazu müssen wir eine zweite ListExploder durchführen, indem wir die Liste Surfaces{ } auswählen.

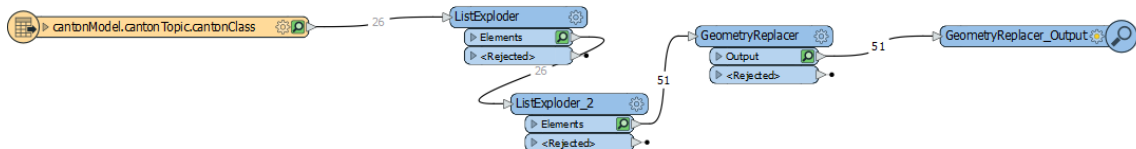


7. Auf diese Weise wird das Attribut Surface freigelegt. Wir stellen fest, dass die Geometrie beim Lesen kodiert wurde. Wir werden den Transformer GeometryReplacer verwenden, um dieses Attribut zu dekodieren und die Geometrie zu erstellen. Überprüfen Sie, ob das in diesem Transformer definierte Encoding mit dem Encoding übereinstimmt, das in den Einstellungen des INTERLIS-Readers definiert ist.



Durch das Anschliessen eines Inspektors können wir nun die Geometrie der Objekte anzeigen. Von diesem Punkt aus können wir zahlreiche Transformer verwenden, um diese Geometrie zu charakterisieren (AreaCalculator, CircularityCalculator, etc...).

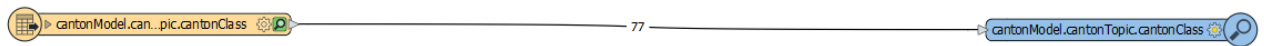
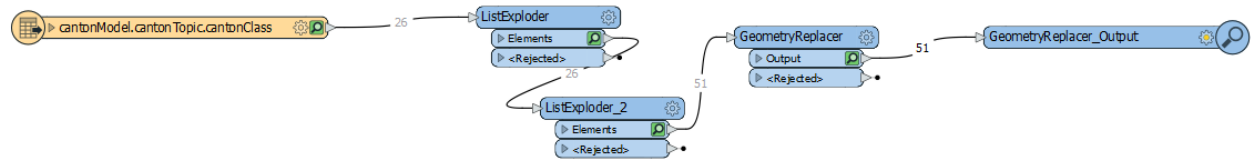
Der Workspace sollte wie folgt aussehen:



Hinweis: Dieses Beispiel ermöglicht es, die INTERLIS-Datendatei nach ihrem Modell zu lesen. Jede Liste entspricht einer Ebene der Verschachtelung der Strukturen. Dieses Konzept der Ebene ist vor allem beim Schreiben einer INTERLIS-Datei wichtig.

8. In diesem Beispiel hätte das Lesen der Datei durch die Verwendung eines bestimmten Attributs vereinfacht werden können. Fügen Sie im selben Workspace einen INTERLIS-Reader hinzu, indem Sie diesmal RepeatFeature in der Einstellung Mapping of multiple Geometry Attribute angeben.

Vergessen Sie nicht, Validate zu deaktivieren. Sehen Sie einen Unterschied zu dem Ergebnis, das Sie erhalten, wenn Sie die Operation manuell durchführen? Wie lässt sich das erklären?



## 2. ÜBUNG 2 - INTERLIS 2 DATEN SCHREIBEN

### 2.1 Einführung

Das Ziel dieser Übung ist es, die ersten 100 Features des Datensatzes "Haltestellen" in INTERLIS 2 zu schreiben.

### 2.2 Daten

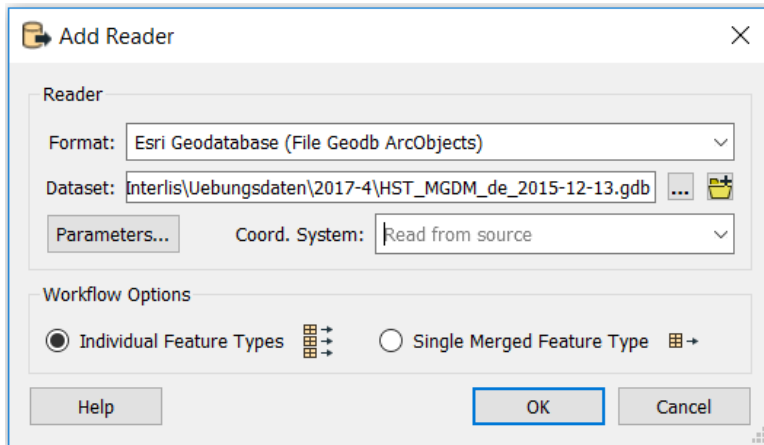
Als Input ist eine GDB (*Ex\_5\_ecriture\_IL2\HST\_MGDM\_de\_2015-12-13.gdb*) verfügbar.

Die Zieldatenstruktur ist ein INTERLIS-2-Modell: *HaltestellenOeV\_V1\_2.ili*

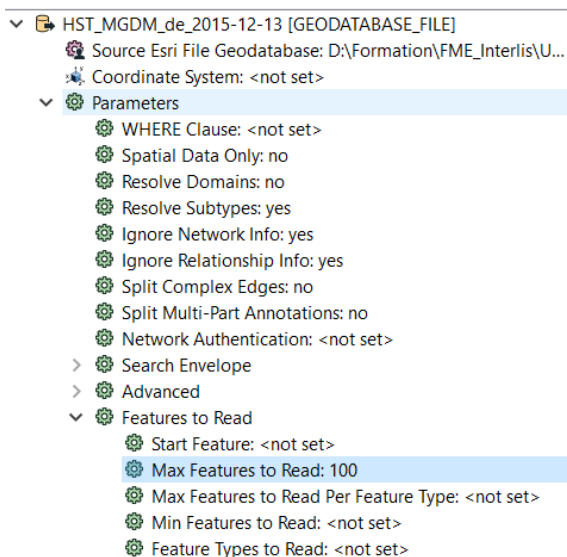
### 2.3 Entwicklung des Workspace

Einen neuen Workspace öffnen und einen neuen Reader hinzufügen :

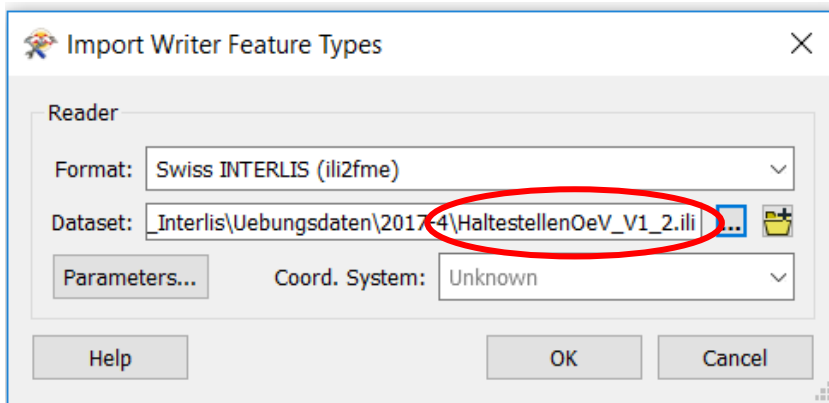
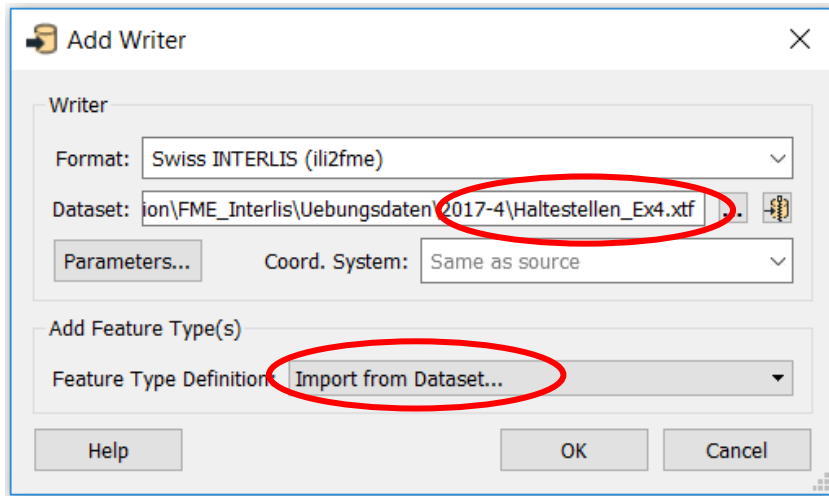
- Format: GBD-Datei
- Datensatz: Betriebspunkt



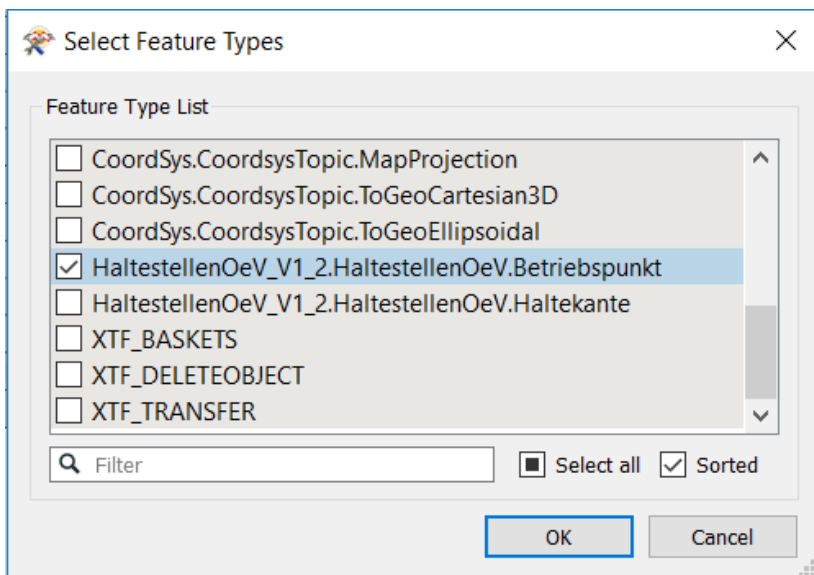
Da wir nur die ersten 100 Features verarbeiten wollen, werden nur diese vom Reader gelesen :



### 2.3.1 Definition des INTERLIS 2 Writers:

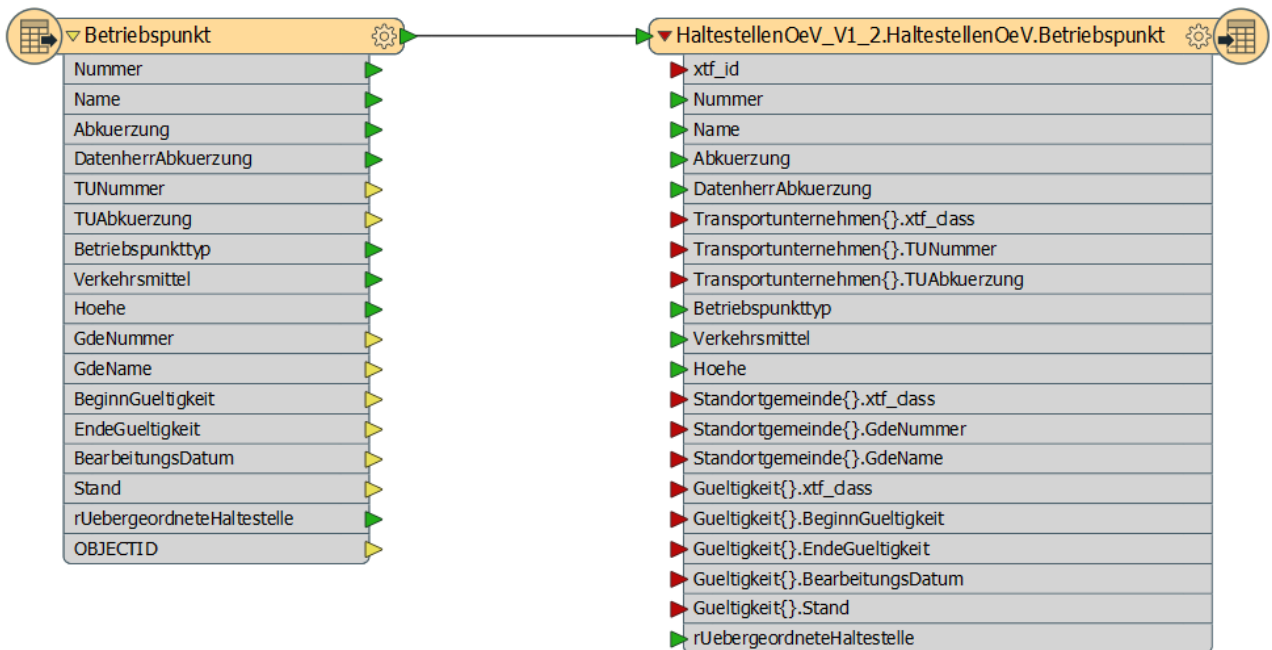


Lesen Sie nur den Feature Type "Betriebspunkt".



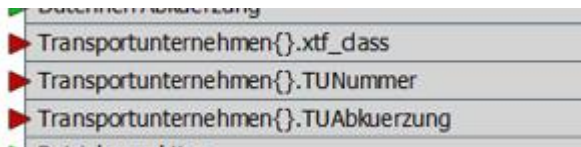


Nun müssen Sie den Reader und den Writer miteinander verbinden.



Wir stellen fest, dass die Attribute der beiden Formate vom ili2fme-Plugin nicht auf die gleiche Weise behandelt werden.

Das Attribut "Transportunternehmen" in INTERLIS wird zum Beispiel mithilfe von Listen gebildet (in FME durch {} gekennzeichnet). Die Liste enthält drei Elemente: xtf\_class, TUNummer und TUAbkuerzung.



Dies entspricht typischerweise der Implementierung eines INTERLIS-Konstrukts "STRUCTURE" :

```

STRUCTURE TU =
  TUNummer:    MANDATORY TEXT*6;
  TUAbkuerzung :    TEXT*10;

CLASS Betriebspunkt =
  Nummer :          MANDATORY 0 .. 9999999;
  Name :           MANDATORY TEXT*50;
  Abkuerzung :      TEXT*6;
  DatenherrAbkuerzung : TEXT*10;
  Transportunternehmen : MANDATORY TU;
  Betriebspunkttyp : MANDATORY Betriebspunkttyp;
  Verkehrsmittel :  Verkehrsmittel;
  Geometrie :       MANDATORY GeometryCHLV03_V1.Coord2;
  Hoehe :           MANDATORY Meereshoehe;
  Standortgemeinde : MANDATORY Gemeinde;
  Gueltigkeit :     MANDATORY Gueltigkeit;
  
```

In diesem Fall wird nur ein Element pro STRUCTURE definiert (eine STRUCTURE könnte mehrere Elemente enthalten).

Dies erleichtert die Implementierung, die mithilfe eines einzigen Transformers möglich sein wird.

Mithilfe eines AttributeManagers :

- Erstellen Sie drei neue Attribute, die in den Quelldaten nicht vorkommen :

Transportunternehmen{0}.xtf_class	<input type="checkbox"/> HaltestellenOeV_V1_2.HaltestellenOeV.TU	Set Value
Standortgemeinde{0}.xtf_class	<input type="checkbox"/> HaltestellenOeV_V1_2.HaltestellenOeV.Gemeinde	Set Value
Gueltigkeit{0}.xtf_class	<input type="checkbox"/> HaltestellenOeV_V1_2.HaltestellenOeV.Gueltigkeit	Set Value

Für den Wert (Attribute Value) verwenden Sie den Wert des STRUCTURE-Namens (vollständig mit Model.Topic.Class).

Die Werte können von hier aus kopiert werden :

*HaltestellenOeV\_V1\_2.HaltestellenOeV.TU*

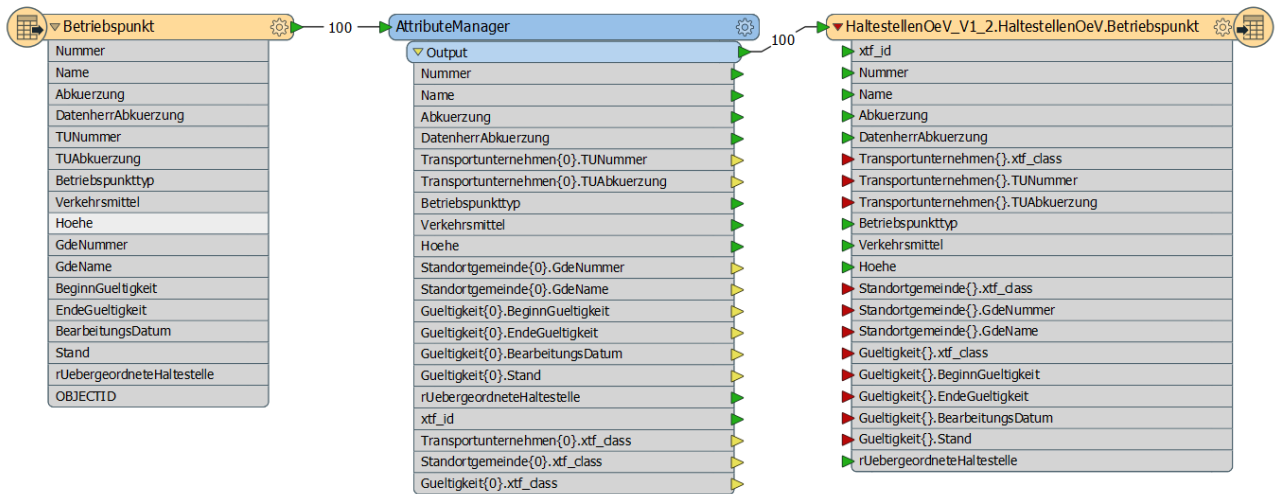
*HaltestellenOeV\_V1\_2.HaltestellenOeV.Gemeinde*

*HaltestellenOeV\_V1\_2.HaltestellenOeV.Gueltigkeit*

- Erstellen Sie eine eindeutige Kennung xtf\_id. Übernehmen Sie dazu einfach die OBJECTID.
- Verknüpfen Sie die Attribute wie folgt :

Input Attribute	Output Attribute	Attribute Value	Action
Nummer	Nummer		Do Nothing
Name	Name		Do Nothing
Abkuerzung	Abkuerzung		Do Nothing
DatenherrAbkuerzung	DatenherrAbkuerzung		Do Nothing
TUNummer	Transportunternehmen{0}.TUNummer		Rename
TUAbkuerzung	Transportunternehmen{0}.TUAbkuerzung		Rename
Betriebspunkttyp	Betriebspunkttyp		Do Nothing
Verkehrsmittel	Verkehrsmittel		Do Nothing
Hoehe	Hoehe		Do Nothing
GdeNummer	Standortgemeinde{0}.GdeNummer		Rename
GdeName	Standortgemeinde{0}.GdeName		Rename
BeginnGueltigkeit	Gueltigkeit{0}.BeginnGueltigkeit		Rename
EndeGueltigkeit	Gueltigkeit{0}.EndeGueltigkeit		Rename
BearbeitungsDatum	Gueltigkeit{0}.BearbeitungsDatum		Rename
Stand	Gueltigkeit{0}.Stand		Rename
rUebergeordneteHaltestelle	rUebergeordneteHaltestelle		Do Nothing
OBJECTID	xtf_id		Rename
	Transportunternehmen{0}.xtf_class	<input type="checkbox"/> HaltestellenOeV_V1_2.HaltestellenOeV.TU	Set Value
	Standortgemeinde{0}.xtf_class	<input type="checkbox"/> HaltestellenOeV_V1_2.HaltestellenOeV.Gemeinde	Set Value
	Gueltigkeit{0}.xtf_class	<input type="checkbox"/> HaltestellenOeV_V1_2.HaltestellenOeV.Gueltigkeit	Set Value

Der Workspace sieht nun so aus :



Stellen Sie den Parameter "Validate" auf No ein.

Versuchen Sie, den Workspace auszuführen und schauen Sie sich das Ergebnis an, indem Sie die XTF mit einem Texteditor öffnen.

```
</HEADERSECTION>
<DATASECTION>
  <HaltestellenOeV_V1_2.HaltestellenOeV BID="101">
    <HaltestellenOeV_V1_2.HaltestellenOeV.Betriebspunkt TID="1">
      <Nnummer>8572922</Nnummer>
      <Name>Wolfenschiessen, Schürmatt</Name>
      <Transportunternehmen>
        <HaltestellenOeV_V1_2.HaltestellenOeV.TU>
          <TUNummer>7</TUNummer>
          <TUAbkuerzung>PAG</TUAbkuerzung>
        </HaltestellenOeV_V1_2.HaltestellenOeV.TU>
      </Transportunternehmen>
      <Betriebspunkttyp>Haltestelle</Betriebspunkttyp>
      <Verkehrsmittel>Bus</Verkehrsmittel>
      <Geometrie>
        <COORD>
          <C1>674215.0</C1>
          <C2>193984.0</C2>
        </COORD>
      </Geometrie>
      <Hoehe>835</Hoehe>
      <Standortgemeinde>
```

Diese Methode ist eine einfache Implementierung von FME-Listen, die in diesem Fall nur aus einem einzigen Element bestehen.

### 3. ÜBUNG 3 – MGDM DATEN LESEN UND GDB EXPORT

#### 3.1 Einführung

Ich habe eine INTERLIS 2.3 Datendatei gemäss MGDM 190.1 (Gewässerraum) erhalten. Ich möchte diese Datei in eine GDB exportieren, um die Daten in mein ArcGIS Pro-Projekt zu integrieren.

#### 3.2 Ziele

- Verwendung von FME, um eine XTF zu lesen und in eine GDB zu exportieren
- Berücksichtigung von Attributen des Typs STRUCTURE

#### 3.3 Daten des Geschäftsjahres

Die Datei *ID190\_1\_Espace\_reserve\_aux\_eaux\_V1.1.xtf* stammt aus einem Export des Kantons Jura, der auf geodienste.ch durchgeführt wurde.

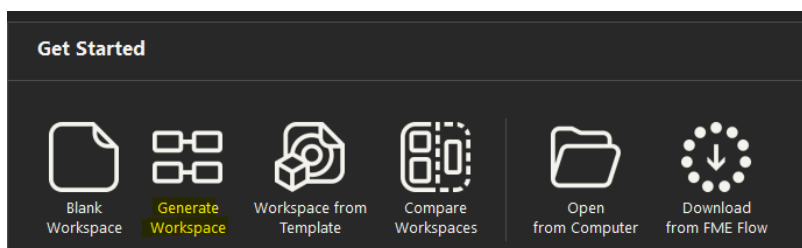
#### 3.4 Vorgehensweise

1. Zunächst schauen wir uns an, wie die ILI-Datei des entsprechenden Modell Gewässerraum\_V1\_1.ili aussieht.

Nachdem Sie die Datei in einem Texteditor geöffnet haben, beantworten Sie die folgenden Fragen:

- Wie viele CLASS enthält das Modell?
- Welche CLASS enthält eine Geometrie? Um welche Art von Geometrie handelt es sich?
- Welche Attribute sind mit einem DOMAIN verbunden?
- Was ist die Kardinalität der Assoziation DokumentGewR?
- Welche vier Attribute sind vom Typ mehrsprachiger Text (MultilingualText)?

2. Öffnen Sie die FME Workbench und klicken Sie auf "Generate Workspace".



3. Für den Reader wählen Sie das INTERLIS-Format und die XTF-Datei, die sich im Ordner der Übung befindet.

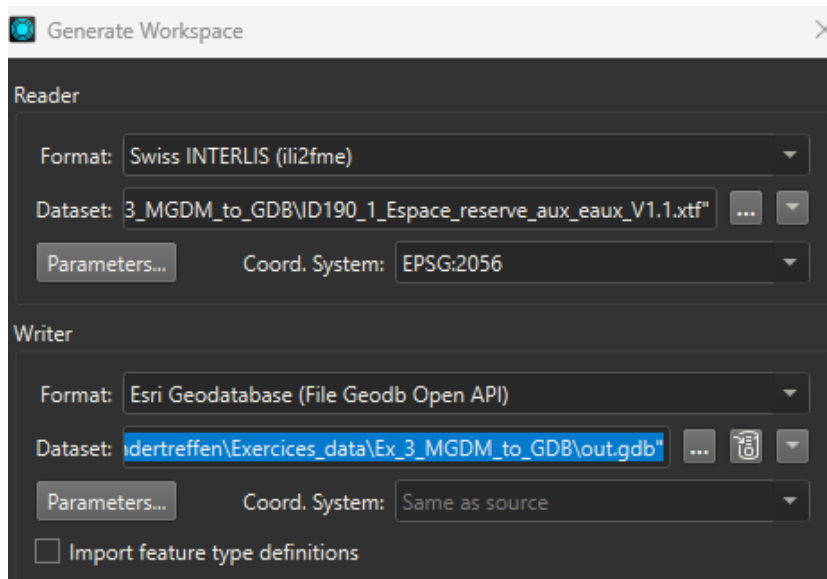
In den Reader-Einstellungen muss der Parameter Validate auf No gesetzt werden. Damit wird überprüft, ob die XTF-Datei beim Lesen mit der entsprechenden ILI-Modell konform ist. In unserem Fall ist die Datei konform und es ist aus Leistungsgründen besser, diesen Parameter zu deaktivieren.



Ausserdem muss das Koordinatensystem des Readers festgelegt werden: LV95, EPSG:2056.

Zu dem Writer wählen Sie das GDB-Format (Open API) und definieren eine Ausgabe-GDB "out.gdb" im Übungsordner.

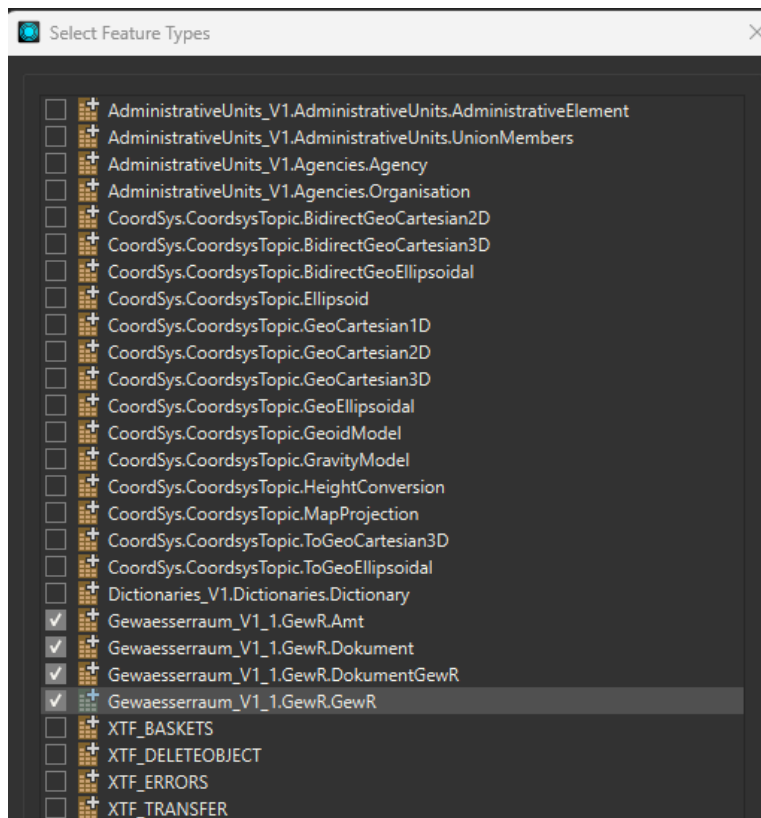
Das Fenster sollte wie folgt aussehen:



4. Klicken Sie auf OK.

Das Fenster "Select Feature Types" wird geöffnet. Es enthält alle Klassen, die sich in unserem Modell "Gewaesserraum\_V1\_1" befinden, sowie in allen Vorlagen, die in diese ILI importiert werden.

In unserem Fall interessieren wir uns nur für die vier Klassen, die aus unserem Modell "Gewaesserraum\_V1\_1" stammen.



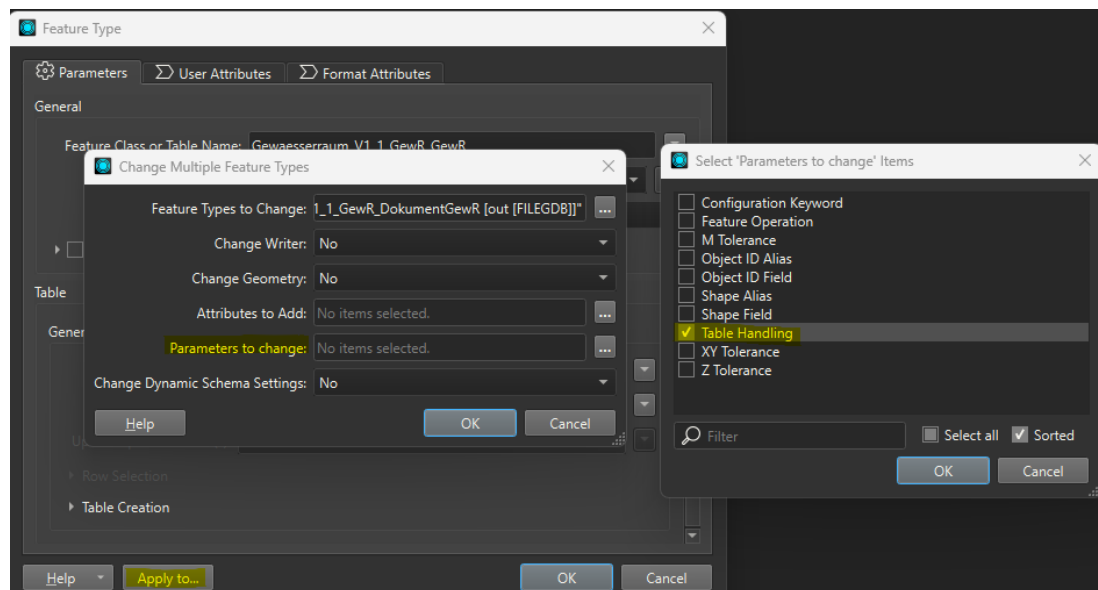
Wir sehen, dass das ili2fme-Plugin uns auch die Klasse DokumentGewR anbietet. Wie bereits erwähnt, handelt es sich dabei um eine Assoziation mit der Kardinalität  $n : m$ , die also durch eine Beziehungstabelle materialisiert wird.

5. Wir erhalten das folgende Fenster :



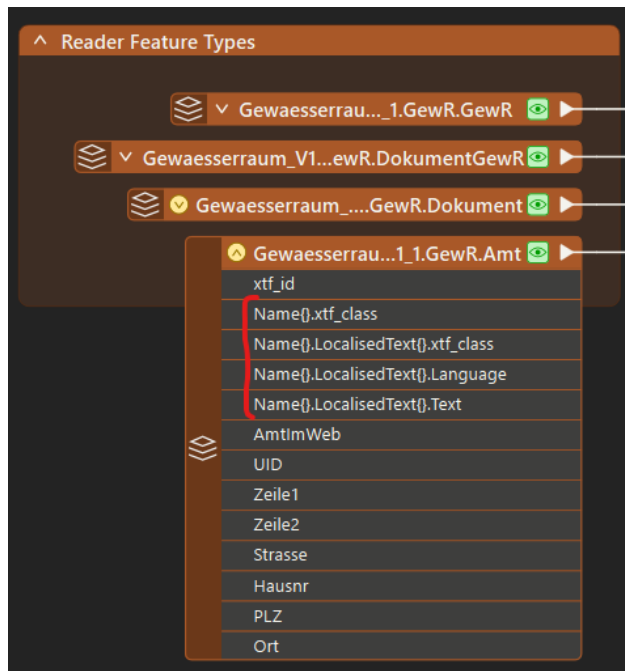
Wenn Sie den Generate Workspace verwenden, haben Sie den Vorteil, dass FME automatisch die entsprechenden Klassen im Writer erstellt und das Mapping durchführt.

6. Klicken Sie auf den FeatureType GewR und stellen Sie "Table Handling" = "Drop and Create" ein. Um diese Änderung auf alle Feature Types anzuwenden, klicken Sie unten im Fenster auf "Apply to...". Wählen Sie unter "Parameters to change" die Option "Table Handling". Klicken Sie dann dreimal auf OK. Wenn eine Bestätigung verlangt wird, klicken Sie auf Yes.



Dadurch werden die Klassen in der GDB bei jeder Ausführung des Workspace neu erstellt. Führen Sie den Workspace aus.

7. Klicken Sie auf den Pfeil, um die Liste der Attribute der Klasse Amt zu öffnen.



Man sieht, dass das Attribut "Name", das in der ILI als mehrsprachiger Text definiert ist, in FME eine Liste ist. Man sieht auch, dass es eine zweite Ebene der Liste gibt: "LocalisedText".

Klicken Sie auf das Auge, um die Daten des Feature Type "Amt" anzuzeigen:

Property	Value
<b>Exposed Attributes (12)</b>	
xtf_id	b63a700d-da50-4bf5-b4aa-1e9048767aab
AmtImWeb	https://www.jura.ch/sdt
Strasse	Rue du 24-Septembre
Hausnr	2
PLZ	2800
Ort	Delémont
<b>Name() (1)</b>	
<b>Name(0)</b>	
xtf_class	LocalisationCH_V1.MultilingualText
<b>LocalisedText() (2)</b>	
<b>LocalisedText(0)</b>	
xtf_class	LocalisationCH_V1.LocalisedText
Text	Service du développement territorial (SDT)
<b>LocalisedText(1)</b>	
xtf_class	LocalisationCH_V1.LocalisedText
Language	fr
Text	Service du développement territorial (SDT)

Wenn wir uns die Struktur des Attributs "Name" genauer ansehen, sehen wir, dass die Liste "Name" ein einziges Objekt enthält (hier in gelb). Dann enthält die Liste "LocalisedText" zwei Objekte (hier in grün).

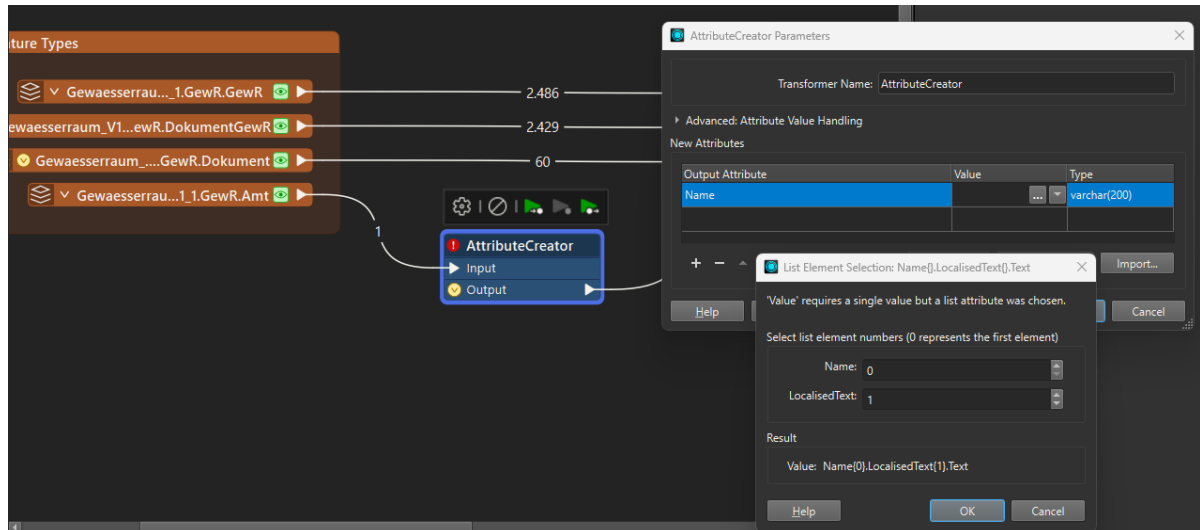
Das erste Element (Index 0) ist generisch. Das zweite (Index 1) enthält den französischen Wert, der hier mit dem generischen Wert identisch ist, nämlich "Service du développement territorial (SDT)". Diese Struktur könnte es ermöglichen, bei Bedarf weitere Elemente in die Liste aufzunehmen, um den Namen in andere Sprachen zu integrieren.

- Wir werden nun ein Attribut "Name" erstellen, das den deutschen Wert dieses mehrsprachigen Attributs enthält.

Fügen Sie nach dem Reader Feature Type "Amt" einen AttributeCreator hinzu.

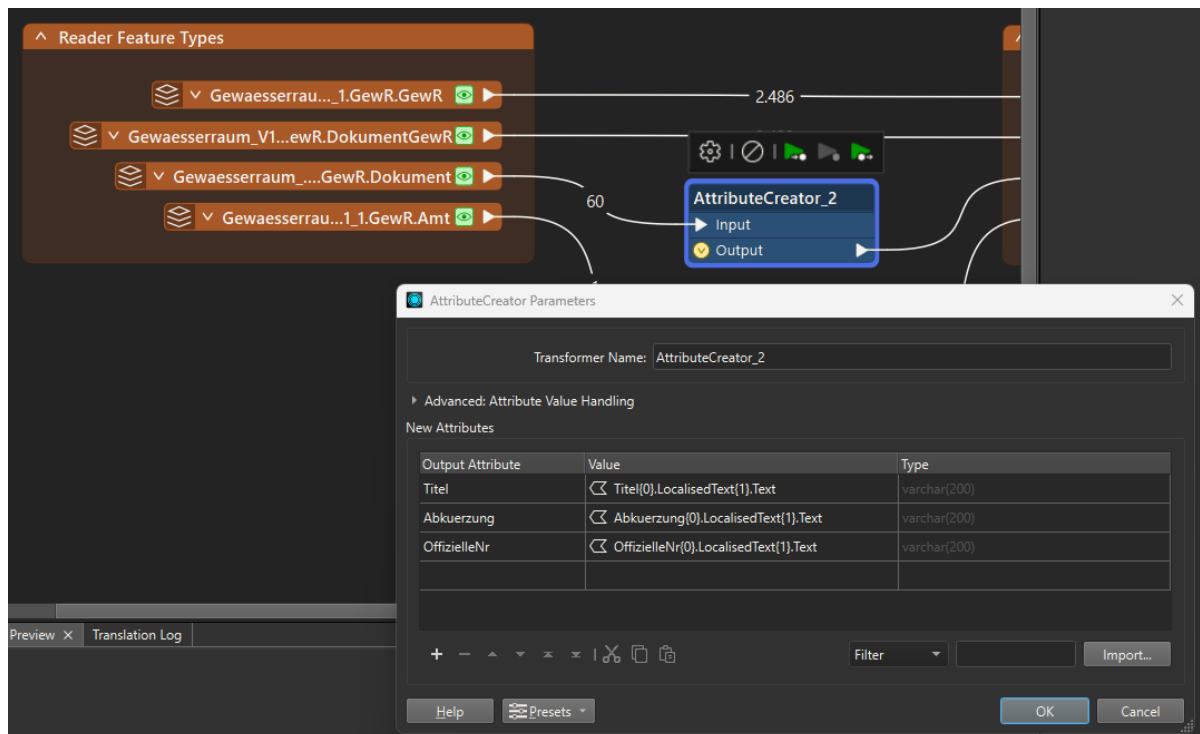
Erstellen Sie ein neues Attribut "Name". Als Wert wählen Sie das Attribut "Name{}.LocalisedText{}.Text".

Da wir in diesen beiden Listen nach einem bestimmten Element suchen wollen, müssen wir angeben, welcher Index aus jeder Liste verwendet werden soll. Wie bereits gesehen, definieren wir 0 und 1 :



Klicken Sie zweimal auf OK.

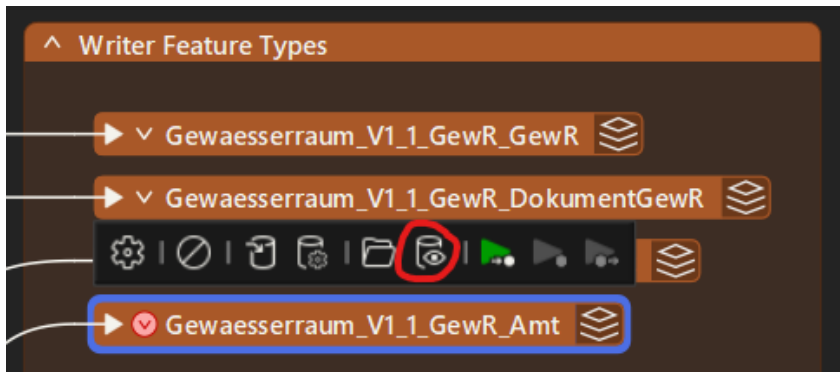
9. Führen Sie das gleiche Verfahren für die drei Attribute Titel, Abkürzung und OffizielleNr der Klasse Dokument durch:





10. Führen Sie den Workspace aus.

Führen Sie einen Rechtsklick auf den Writer Feature Type aus und klicken Sie dann auf den Knopf unten (rot markiert), um die geschriebenen Daten anzuzeigen.



In "Amt" und "Dokument" sind die betreffenden Attribute sichtbar, wobei der Wert dem französischen Wert des mehrsprachigen Attributs entspricht.

Wenn man sich die GDB in der Ausgabe mit ArcGIS Pro ansieht, wurden alle vier Klassen korrekt erstellt:



## 4. ÜBUNG - OPTIONAL - MGDM-EXPORT MIT FME

### 4.1 Einführung

Ausgehend von einer GDB, die Daten des Kantons Freiburg enthält, besteht das Ziel der Übung darin, diese Daten mit FME in das entsprechende MGDM zu exportieren.

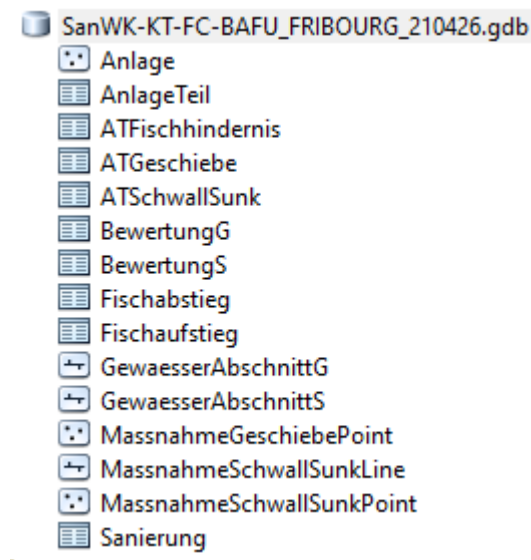
Die Vorlage lautet *SanierungWasserkraft\_LV03\_V1\_2* (ID 192).

### 4.2 Ziele

- Verwendung von FME zum Exportieren einer GDB in INTERLIS
- Überprüfen Sie das Ergebnis mit iliValidator

### 4.3 Vorgehensweise

Die Eingabe-GDB enthält die folgenden Tabellen, die alle Objekte enthalten :



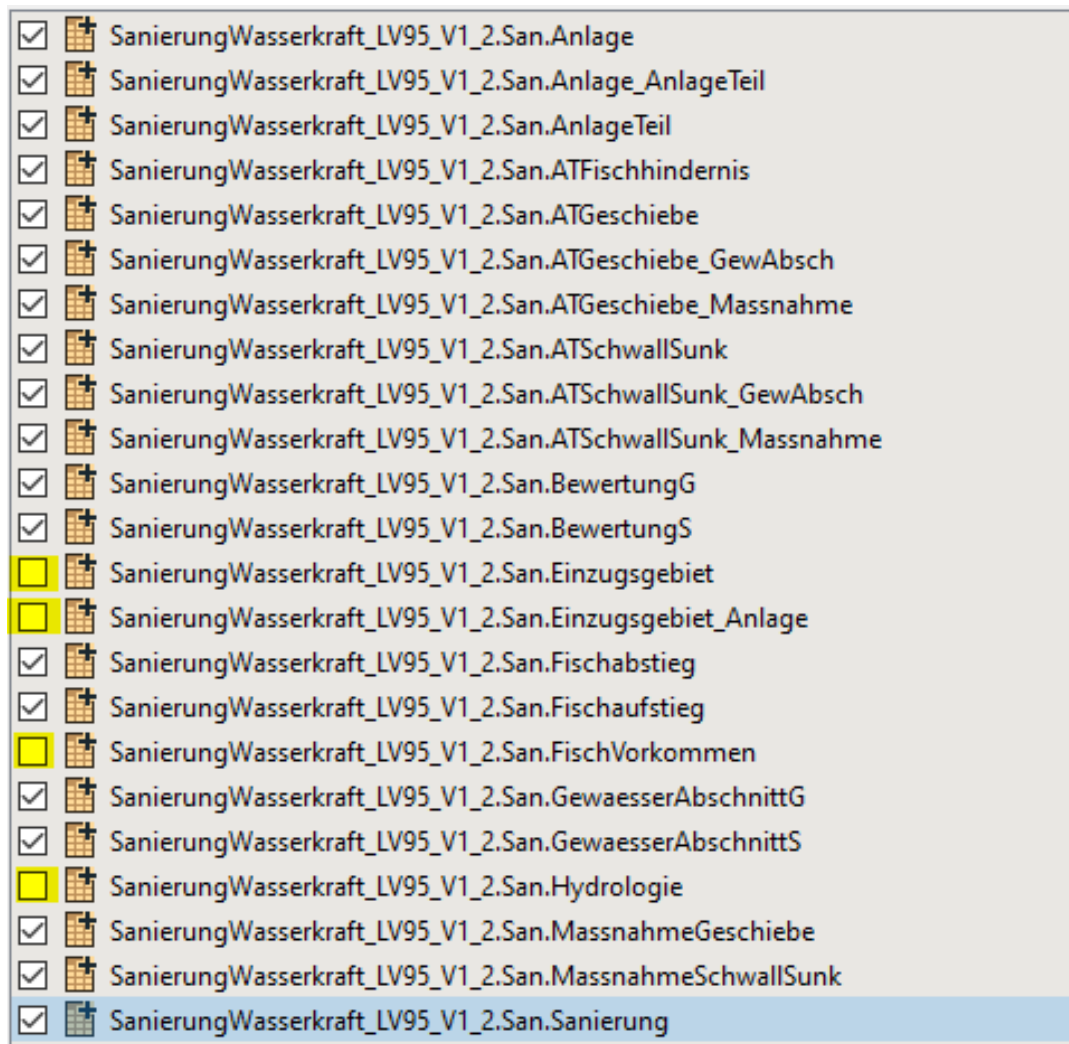
Diese Tabellen kommen der Modellierung nach dem ILI bereits sehr nahe.

1. Einen leeren Workspace öffnen
2. Fügen Sie einen OpenAPI GDB-Reader hinzu, indem Sie alle Tabellen der GDB laden. Die Datenbank befindet sich im Ordner GDB\_IN

Bevor wir in das Thema einsteigen, ist es wichtig, sich in das INTERLIS-Modell "einzufühlen":

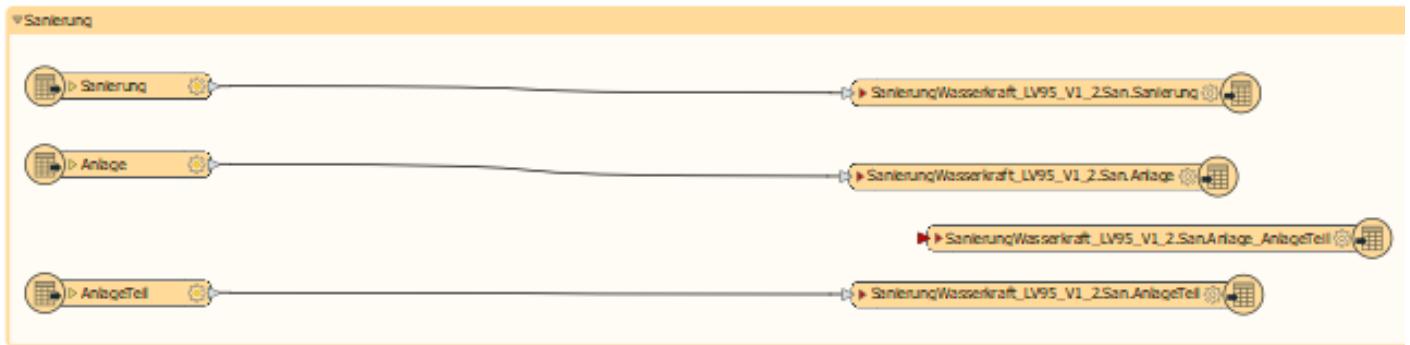
- Nachdem Sie das ILI (SanierungWasserkraft\_V1\_2.ili im ILI\_OUT-Ordner) mit Notepad++ geöffnet haben :
  - Die verschiedene Modelle (MODEL) der Datei markieren
  - Den/die TOPIC(s) jedes Modells ausfindig machen
  - DOMAINS und STRUCTURE finden und schauen, wo sie verwendet werden
  - Die verschiedenen CLASSES und die zwischen ihnen bestehenden ASSOCIATIONS ausfindig machen (es gibt viele davon...).
  - Welche Klassen enthalten Geometrie?
  - Welcher Art von Geometrie entspricht die "StrOrt"-Geometrie?

- Öffnen Sie das PDF "190206\_Planung\_und\_Berichte... .pdf", das sich im selben Ordner befindet.
  - Schlagen Sie auf Seite 19 das UML-Diagramm auf: Versuchen Sie, einige Klassen/Assoziationen mit dem ILI-Modell in Verbindung zu bringen.
  - Markieren Sie den Bereich im Diagramm, der sich auf :
    - Sanierung Wasserkraft
    - Geschiebehaushalt
    - Schwall-Sunk
    - Fischwanderung
  - Beziehen Sie sich auf die Seiten 12-13 der PDF, um in groben Zügen zu verstehen, worauf sich diese Begriffe beziehen.
- 3. Fügen Sie einen INTERLIS-Writer hinzu, indem Sie die folgenden Klassen aus dem ILI-Modell laden (Sie müssen keine Klassen importieren, die sich nicht in der GDB befinden) :

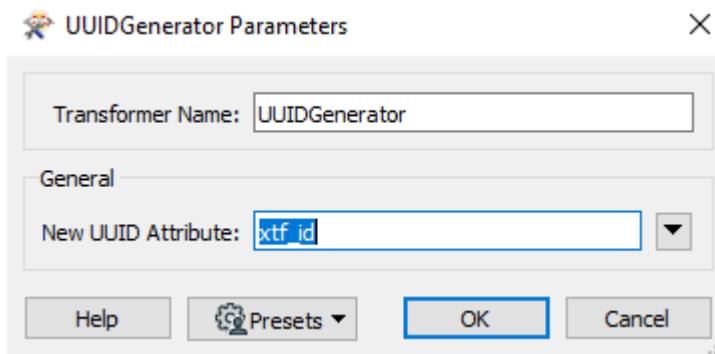


- 4. Beantworten Sie die folgenden Fragen:
  - Was haben die folgenden Klassen des INTERLIS-Modells gemeinsam:  
Anlage\_AnlageTeil / ATGeschiebe\_GewAbsch / ATGeschiebe\_Massnahme /  
ATSchwallSunk\_GewAbsch / ATSchwallSunkk\_Massnahme / Einzugsgebiet\_Anlage?
  - Warum werden sie in FME importiert?

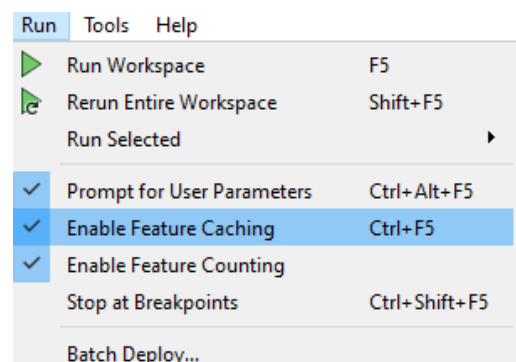
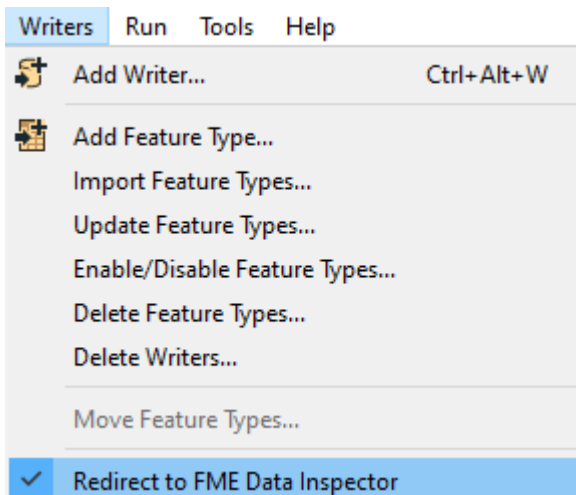
5. Sobald die Klassen importiert sind, konzentrieren wir uns auf den "Sanierungs"-Teil der Vorlage. Formatieren Sie den Workspace so, dass er in etwa so aussieht :



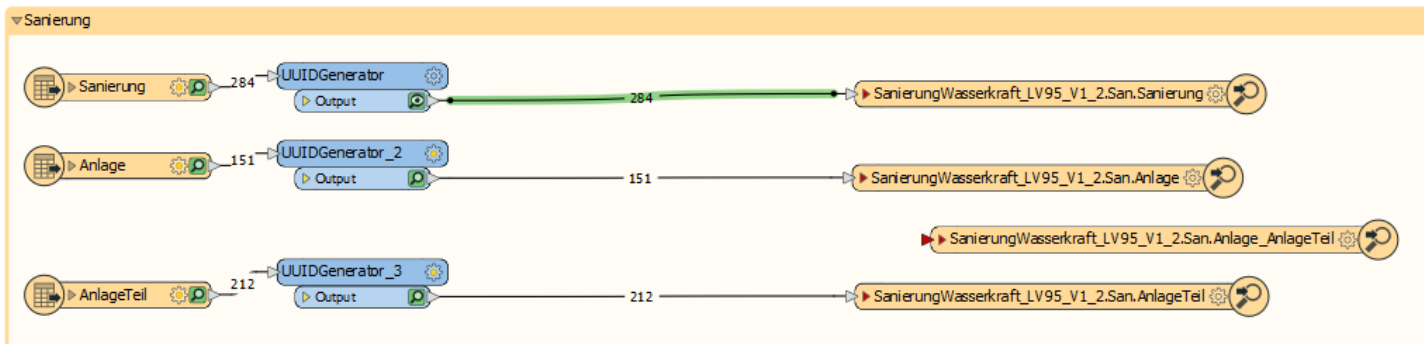
6. Für jede Klasse muss bereits eine xtf\_id zu jedem Objekt erzeugt werden, indem 3 UUIDGenerator :



7. Aktivieren Sie in Writers "Redirect to FME Data Inspector" sowie in Run "Enable Feature Caching". Dies ist bei der Entwicklung praktisch.



8. Führen Sie den Workspace aus, der im Moment so aussieht :



9. In Bezug auf die Klasse Sanierung sehen wir in der ILI, dass das Attribut Phase obligatorisch ist und auf den Katalog SanPhaseRef verweist. Klicken Sie auf die kleine Lupe am Ausgang des ersten UUIDGenerators, um die Daten an dieser Stelle anzuzeigen. Wir können sehen, dass das Attribut Phase einen Wert zwischen 1 und 7 hat. Wenn man in der XML-Datei der Kataloge nach SanPhase sucht, findet man Codes, die wie folgt aussehen:

```
<SanierungWasserkraft_Codelisten_V1_2.Catalogs.SanPhase TID="":
  <ItemId OID="764eb653-5396-4a5c-b2b9-66558ab6d375"/>
  <Code>SanPhase7</Code>
  <Name>
    <LocalisationCH_V1.MultilingualText>
      <LocalisedText>
        <LocalisationCH_V1.LocalisedText>
          <Language>de</Language>
          <Text>Detailplanung (G)</Text>
        </LocalisationCH_V1.LocalisedText>
        <LocalisationCH_V1.LocalisedText>
          <Language>fr</Language>
          <Text>Planification détaillée (C)</Text>
        </LocalisationCH_V1.LocalisedText>
      </LocalisedText>
    </LocalisationCH_V1.MultilingualText>
  </Name>
</SanierungWasserkraft_Codelisten_V1_2.Catalogs.SanPhase TID="":
```

Zum Beispiel hier: "SanPhase7". In diesem Fall ist es die OID, die in den Attributwert eingetragen werden muss. Wir verwenden daher den Transformer AttributeValueMapper, um die Phasennummer auf die entsprechende OID im XML der Kataloge abzubilden. Wir erhalten also :

Transformer Name:

Attribute Selection

Source Attribute:

Destination Attribute:

Default Value:

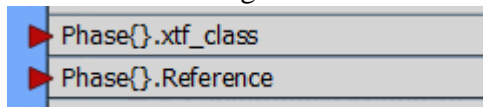
Value Map

Mapping Direction:

Source Value	Destination Value
1	89265257-90d3-480f-9cd2-57cff8c14c28
2	13532b78-72c5-4f3f-bab4-a0f9e7f941b9
3	d14474b4-4596-4f1d-b97e-e638e72c0123
4	ed967817-8323-4f4d-978e-146bd550e361
5	b5fda6d6-61e2-41bc-9dc7-e7feef2169b5
6	e55eebb7-acf0-46c9-816e-0d70a65f00bc
7	764eb653-5396-4a5c-b2b9-66558ab6d375

Buttons: Help, Presets, OK, Cancel, Import...

10. Nun müssen wir das Attribut Phase{ } korrekt ausfüllen, das eine Liste ist, da es eine Interlis-Struktur für Kataloge enthält.



Dazu verwenden wir einen AttributeCreator, direkt nach dem AttributeValueMapper :

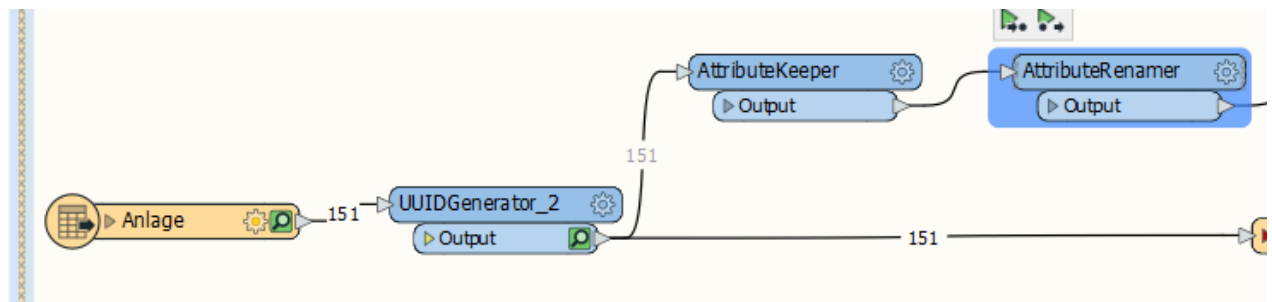
Transformer Name:

> Advanced: Attribute Value Handling

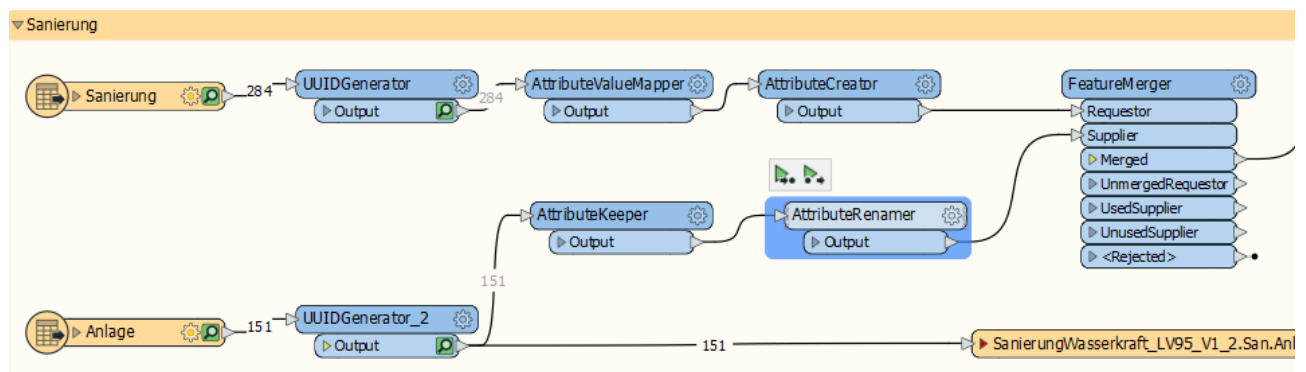
Attributes To Create

New Attribute	Attribute Value
Phase{0}.xtf_class	<input type="checkbox"/> SanierungWasserkraft_Codelisten_V1_2.Catalogs.SanPhaseRef
Phase{0}.Reference	<input checked="" type="checkbox"/> Phase

11. Ebenfalls in der Klasse Sanierung sehen wir, dass der Writer das Attribut rBauwerk enthält. Unter Bezugnahme auf das Modell sehen wir, dass es sich um einen Fremdschlüssel zur Tabelle Anlage handelt. In der Eingabe-GDB sehen wir, dass es sich um das Attribut rSanierung handelt, das eine Verknüpfung zum Attribut AnlageNr der Tabelle Anlage enthält. Da wir eine technische ID als Fremdschlüssel benötigen, verwenden wir einen FeatureMerger, um dieses Attribut abzurufen. Platzieren Sie zunächst einen AttributeKeeper nach dem UUIDGenerator\_2, um nur die Attribute xtf\_id und AnlageNr zu behalten. Platzieren Sie dann einen AttributeRenamer, um xtf\_id in rBauwerk umzubenennen.



Dann fügen Sie den FeatureMerger wie folgt ein:



**FeatureMerger Parameters**

Transformer Name:

☐ Group Processing

Transformer Mode

Suppliers First:

Join On

Requestor	Supplier	Comparison Mode
<input type="checkbox"/> rSanierung	<input type="checkbox"/> AnlageNr	<input type="checkbox"/> Automatic

+ - ✂ 📄 🗑

Merge Parameters

Feature Merge Type:

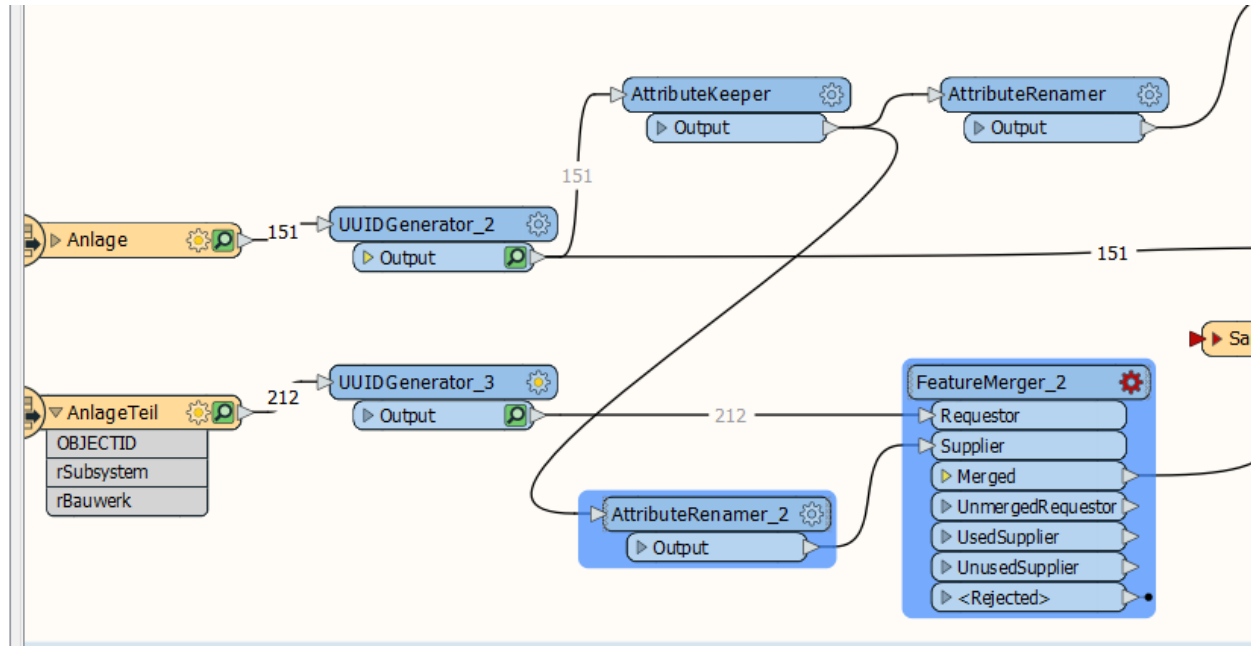
Reject Null and Missing Keys:

☐ Process Duplicate Suppliers

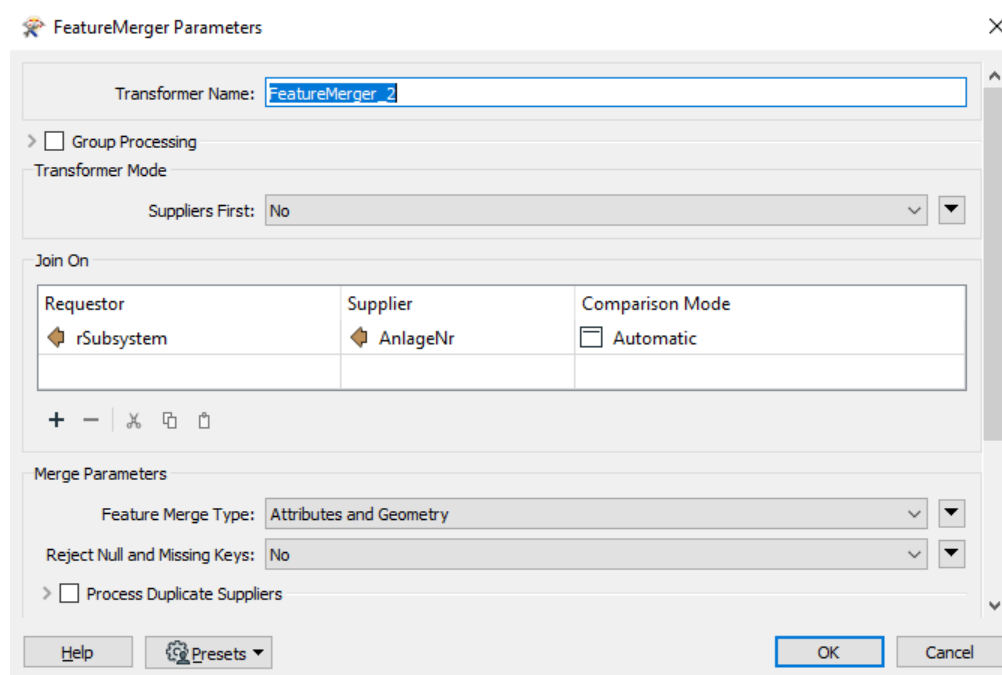
Auf diese Weise ist das Attribut rBauwerk nun korrekt ausgefüllt.

12. Ein ähnlicher Vorgang muss für das Attribut Typ der Klasse Anlage durchgeführt werden, das ebenfalls ein Pflichtattribut ist. Dieser Schritt ist optional.
13. Nun müssen wir noch die Tabelle AnlageTeil und die Beziehungstabelle Anlage\_AnlageTeil ausfüllen. In der GDB sehen wir, dass die Tabelle AnlageTeil zwei Attribute enthält:
  - rSubsystem, das ein Fremdschlüssel auf AnlageNr ist
  - rBauwerk, die FachID von AnlageTeil ist

Wie zuvor müssen wir den technischen Schlüssel von AnlageTeil mithilfe eines FeatureMergers wie folgt suchen: Fügen Sie zunächst einen AttributeRenamer hinzu, um xtf\_id in rKontroller umzubenennen:

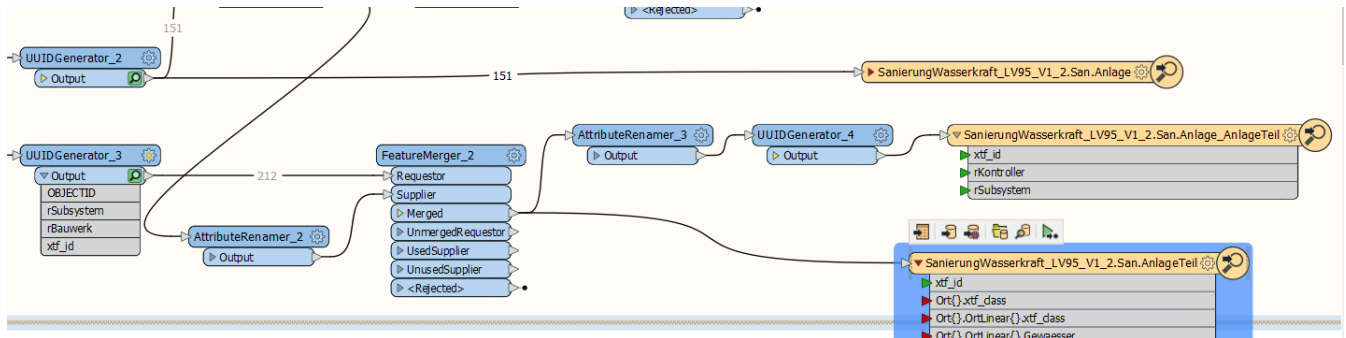


Fügen Sie dann den FeatureMerger hinzu. Vergessen Sie nicht, für den Feature Merge Type "Attributes and Geometry" einzustellen. Die von Anlage abgerufenen Geometrien werden später verwendet.

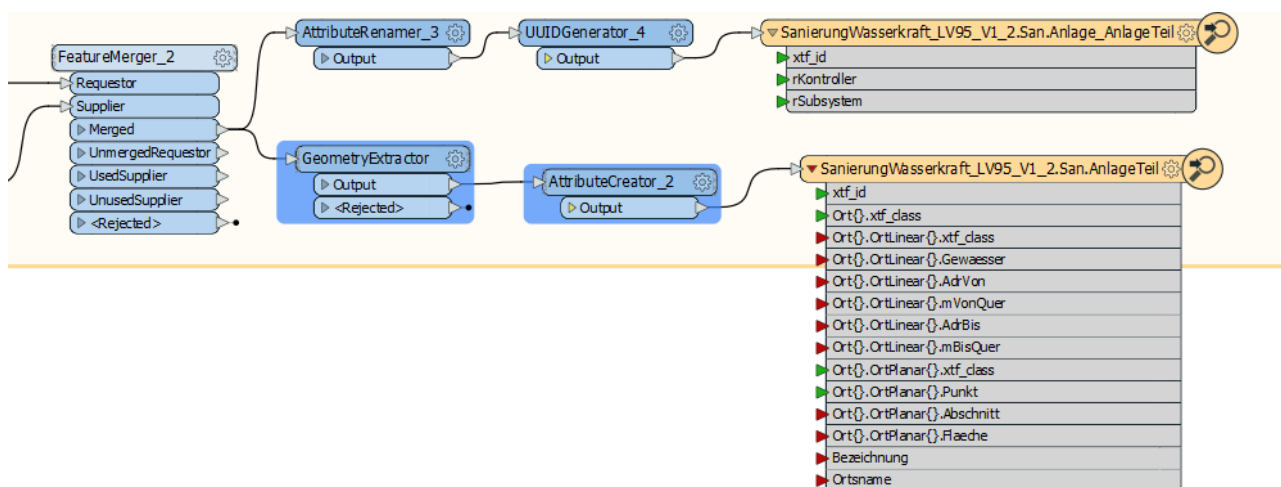




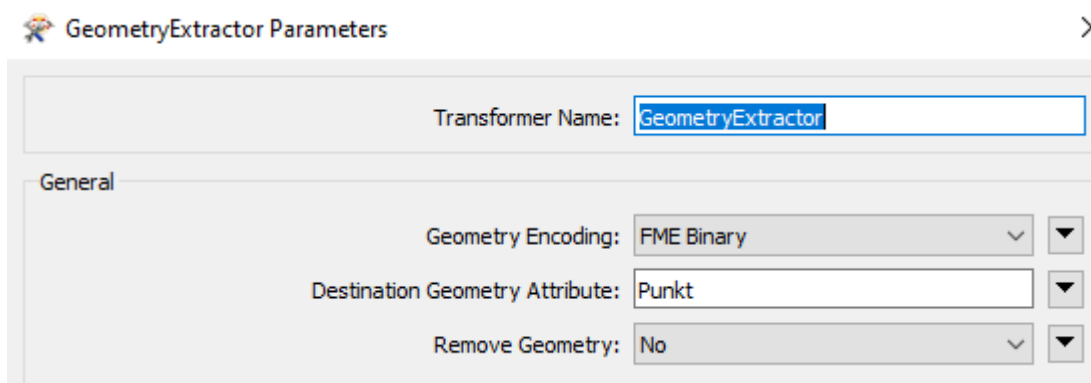
14. Um die Beziehungstabelle Anlage\_AnlageTeil zu füllen, benötigen wir einen AttributeRenamer am Ausgang des FeatureMergers, um rBauwerk in rSubsystem umzubenennen. Wir müssen noch einen UUIDGenerator hinzufügen, um eine xtf\_id für jedes Objekt zu erhalten.

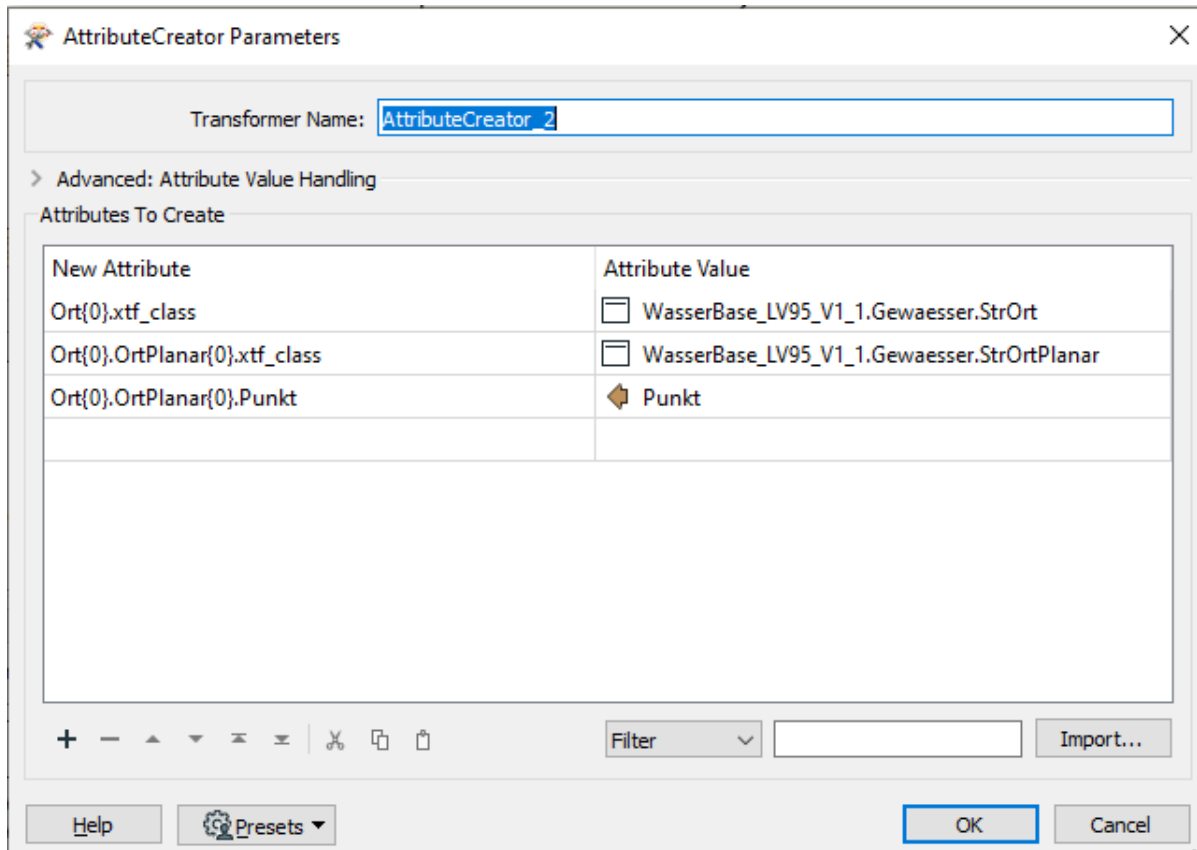


15. Die Interlis-Klasse AnlageTeil enthält eine Geometrie (Attribut Ort{}), die jedoch nicht obligatorisch ist. Für die Übung werden wir dennoch die Geometrie Punkt verwenden, die wir mit dem FeatureMerger abgerufen haben, um eine Geometrie zu rendern. Platzieren Sie zunächst einen GeometryExtractor, dann einen AttributeCreator...



...mit den folgenden Einstellungen:





16. Legen Sie in den Writer-Einstellungen die zu schreibende xtf-Datei fest. Zum Beispiel:  
 /ILI\_OUT/ SanierungWasserkraft\_V1\_2.xtf  
 Deaktivieren Sie Validate im Interlis-Writer.  
 Entfernen Sie den Redirect to Visualizer und versuchen Sie, den Workspace laufen zu lassen.  
 Überprüfen Sie, ob der xtf erstellt wurde und die Struktur korrekt erscheint.
17. Das Ergebnis kann dann an den iliValidator-Checker weitergegeben werden. Eine Version ist bei Bedarf online verfügbar (ilicop.ch). Zu diesem Zeitpunkt existiert eine grosse Anzahl von Fehlern beim Checker. Es beginnt also die Phase, in der die XTF-Datei so lange korrigiert wird, bis sie dem Checker keine Fehler mehr liefert...  
 Versuchen Sie, die Fehler im Log zu verstehen und sie nach Möglichkeit zu beheben.