

4ème journée des utilisateurs INTERLIS

# **Workshop INTERLIS, FME et ESRI**

Exercices



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

Autor	: AV
Version	: 1.0 du 05.11.2024

## TABLE DES MATIERES

<b>1.</b>	<b>EX. 1 - LECTURE D'UN FICHIER INTERLIS 2 .....</b>	<b>3</b>
1.1	Présentation.....	3
1.2	Création du workspace FME .....	3
<b>2.</b>	<b>EX. 2 – ÉCRIRE DES DONNÉES INTERLIS 2 .....</b>	<b>7</b>
2.1	Introduction.....	7
2.2	Données.....	7
2.3	Développement du workspace .....	7
2.3.1	Définition du Writer INTERLIS 2: .....	8
<b>3.</b>	<b>EX. 3 – LECTURE MGDM ET EXPORT EN GDB.....</b>	<b>12</b>
3.1	Introduction.....	12
3.2	Buts .....	12
3.3	Données de l'exercice .....	12
3.4	Marche à suivre.....	12
<b>4.</b>	<b>EX. 4 – FACULTATIF - EXPORT MGDM AVEC FME .....</b>	<b>18</b>
4.1	Introduction.....	18
4.2	Buts .....	18
4.3	Marche à suivre.....	18

## 1. EX. 1 - LECTURE D'UN FICHIER INTERLIS 2

### 1.1 Présentation

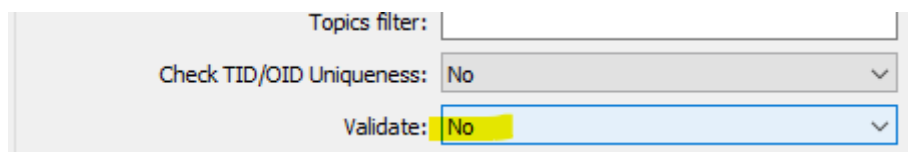
Pour cet exercice, nous disposons d'un fichier INTERLIS 2.3 qui contient des données sur les cantons (données descriptives ainsi que géométriques). Un canton peut être composé de plusieurs parties (forme principale et enclave). La géométrie est donc de type multi-part.

L'objectif est de lire le fichier Interlis et d'aller rechercher la géométrie dans la structure correspondante.

### 1.2 Création du workspace FME

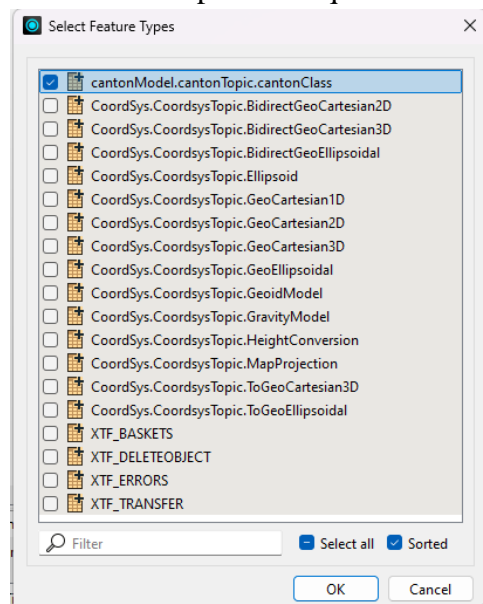
Lecture du fichier Interlis 2.3 :

1. Dans l'onglet Readers, sélectionner Add reader
2. Le format sera du *Swiss INTERLIS (ili2fme)* et il faut sélectionner le fichier *cantons.xtf* comme data set.
3. En cliquant sur le bouton *Parameters*, il faut désactiver la fonction Validate. Pour l'exercice, cela permet de gagner du temps à la lecture du fichier xtf.

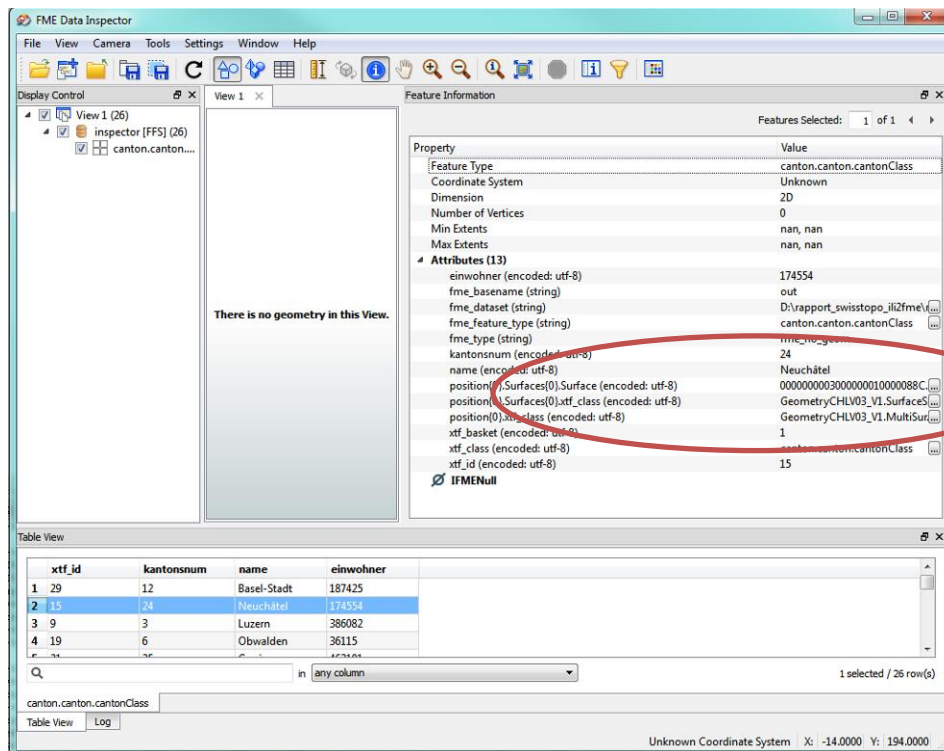


Valider les deux fenêtres en cliquant sur OK pour importer les données

4. FME propose l'import de toutes les classes qui sont présentes dans nos modèles de données (le général, *cantonModel.ili*, ainsi que tous ceux importés via la balise *IMPORT*). Dans le cas présent, nous allons importer uniquement la classe : *cantonClass*.



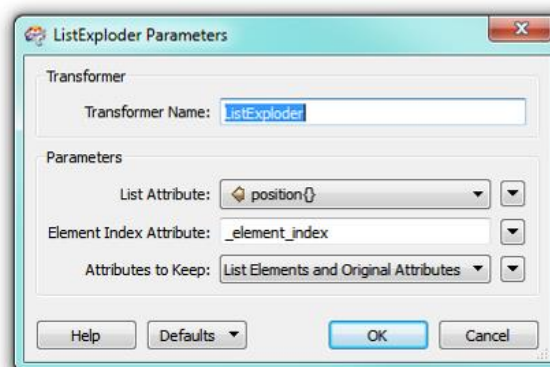
5. En connectant un inspecteur au reader fraîchement importé, nous n'obtenons aucune géométrie. Cependant, nous pouvons constater que quatre attributs sont présents. Le premier attribut correspond à l'*xtf\_id* et les suivants sont les attributs descriptifs des cantons. La géométrie est stockée sous forme de liste. C'est de cette manière qu'ili2fme permet de regrouper des géométries multi-parties. En cliquant sur une valeur des attributs affichés, nous pouvons les voir dans la fenêtre Feature Information.



Afin de pouvoir utiliser ces informations, il faudra donc extraire ces listes.

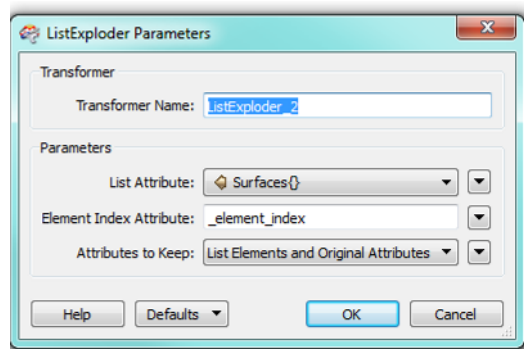
A noter que dans le cas d'un XTF où la géométrie n'est pas multi-part, elle sera directement reconnue par le plugin ili2fme.

6. Puis à l'aide d'un ListExploder, nous allons exposer la liste `position{ }`, ce qui correspond au nom de l'attribut géométrie dans le modèle ili.

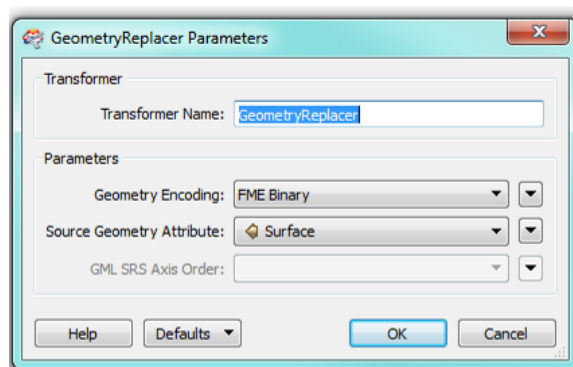


En reconnectant un inspecteur à la sortie de ce ListExploder, nous pouvons voir que l'attribut *xtf\_class* a été créé. Les valeurs de cet attribut correspondent à la classe de géométrie des cantons.

Cependant, il n'est toujours pas possible de visualiser la géométrie. Pour ce faire, nous devons effectuer une seconde exposition de liste, en sélectionnant la liste Surfaces{ }.

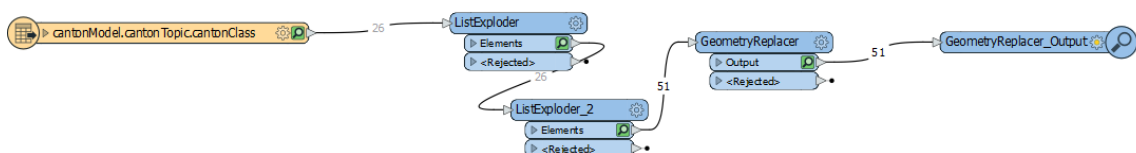


7. Ainsi, l'attribut Surface est exposé. On remarque que la géométrie a été encodée à la lecture. Nous allons utiliser le transformateur GeometryReplacer pour décoder cet attribut et créer la géométrie. Vérifier que l'encoding défini dans ce transformateur corresponde avec l'encoding qui est défini dans les paramètres du reader INTERLIS.



En connectant un inspecteur, nous pouvons maintenant visualiser la géométrie des objets. A partir de ce point, il est possible d'utiliser de nombreux transformateurs pour caractériser cette géométrie (AreaCalculator, CircularityCalculator, etc...).

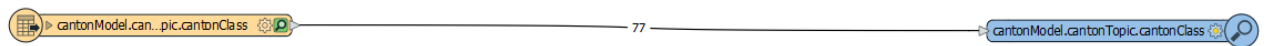
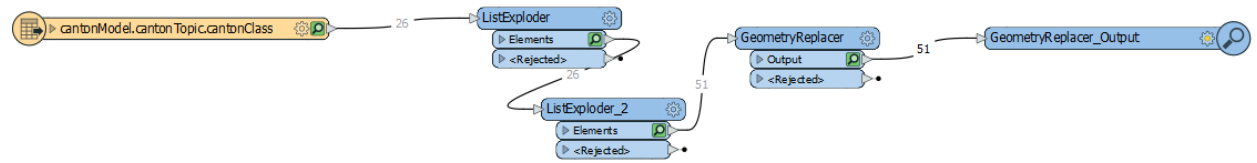
Le workspace doit ressembler à cela :



Remarque: cet exemple permet de lire le fichier de données INTERLIS en fonction de son modèle. Chaque liste correspond à un niveau d'imbrication des structures. Cette notion de niveau est importante principalement lors de l'écriture d'un fichier INTERLIS.

8. Dans cet exemple, la lecture du fichier aurait pu être simplifiée en utilisant un attribut spécifique. Dans le même workspace, ajouter un Reader INTERLIS en spécifiant cette fois RepeatFeature dans le paramètre Mapping of multiple Geometry Attribute. Ne pas oublier de désactiver Validate.

Voyez-vous une différence avec le résultat obtenu en faisant l'opération manuellement ?  
Comment l'expliquer ?



## 2. EX. 2 – ÉCRIRE DES DONNÉES INTERLIS 2

### 2.1 Introduction

L'objectif de cet exercice est d'écrire les 100 premiers Features du jeu de données « Haltestellen » en INTERLIS 2.

### 2.2 Données

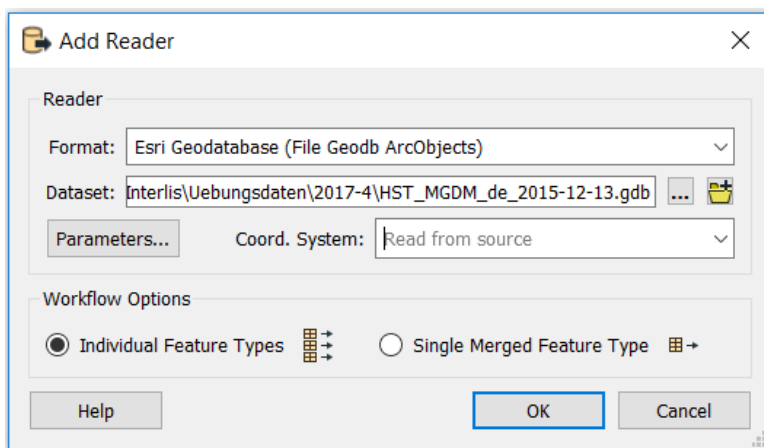
Comme input, une GDB (*Ex\_5\_ecriture\_IL2\HST\_MGDM\_de\_2015-12-13.gdb*) est disponible.

La structure de données cible est un modèle INTERLIS 2 : *HaltestellenOeV\_V1\_2.ili*

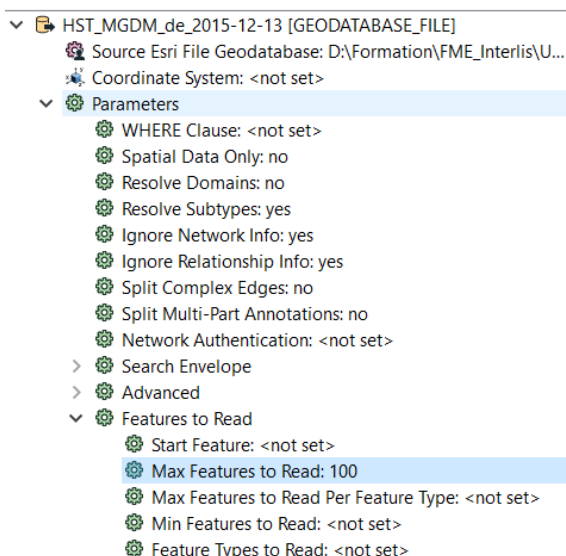
### 2.3 Développement du workspace

Ouvrir un nouveau workspace et ajouter un nouveau Reader :

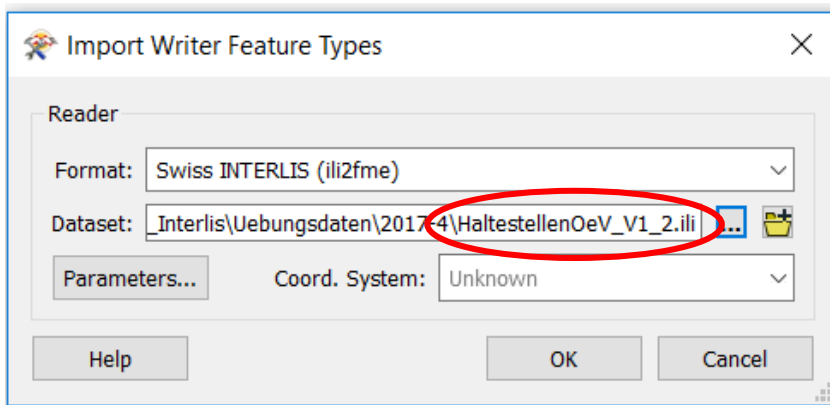
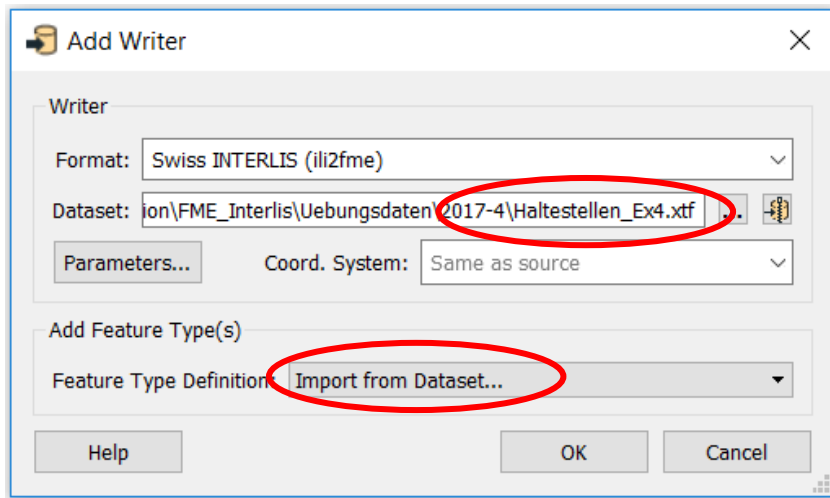
- Format: file GBD
- Jeu de données: Betriebspunkt



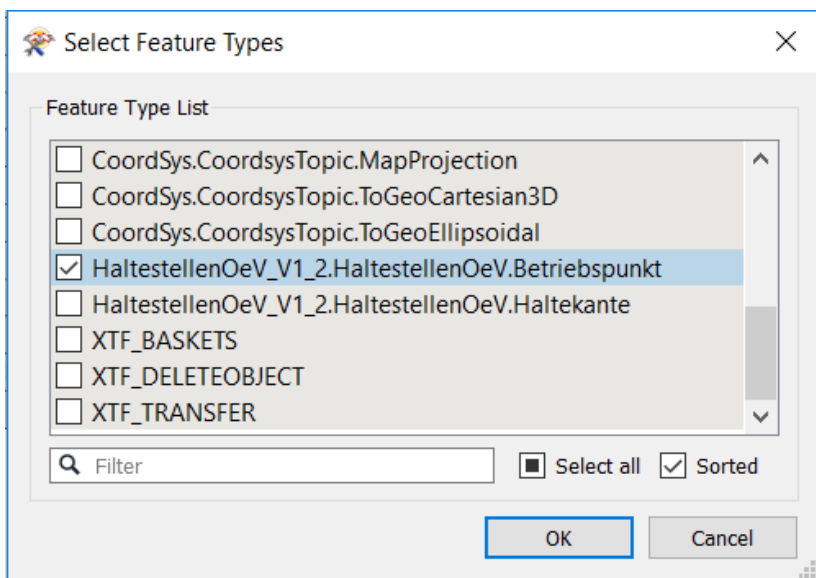
Comme nous ne voulons traiter que les 100 premiers Features, seuls ceux-ci seront lus par le Reader :



### 2.3.1 Définition du Writer INTERLIS 2:

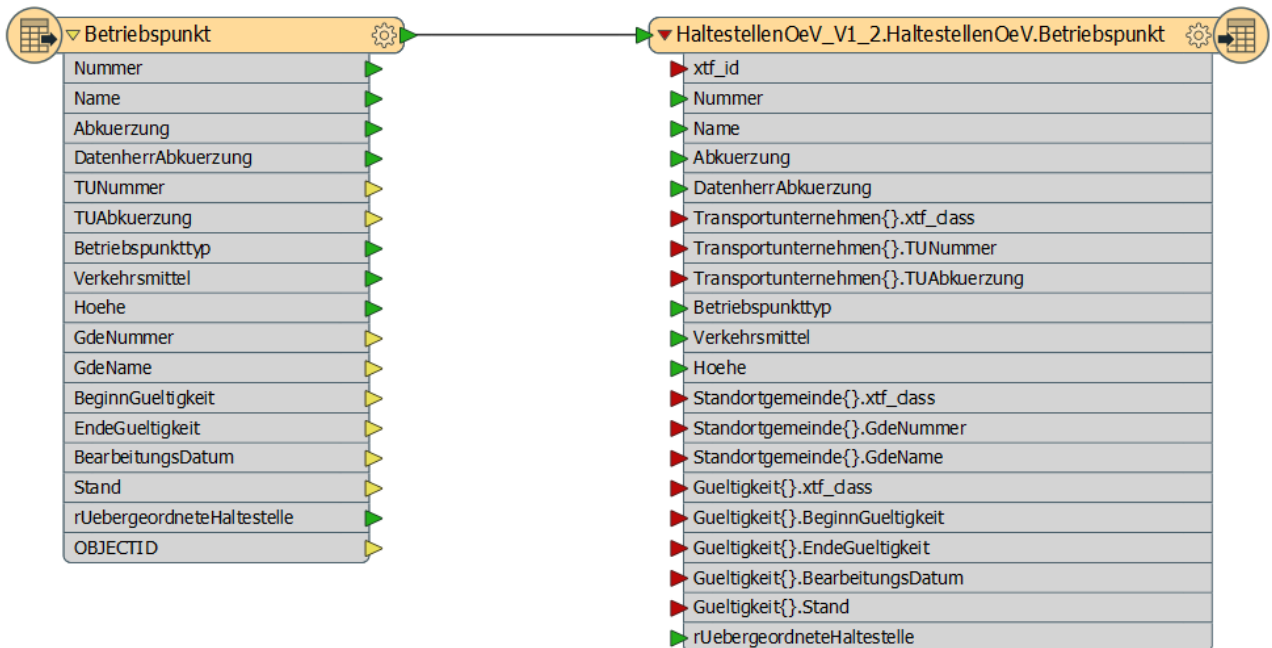


Ne lire que le Feature Type « Betriebspunkt ».



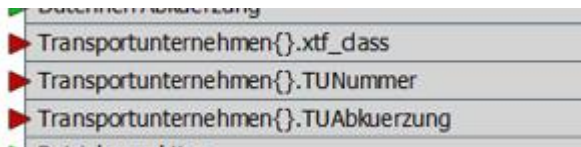


Maintenant, il faut relier le Reader et le Writer.



Nous remarquons que les attributs des deux formats ne sont pas traités de la même manière par le plugin ili2fme.

Par exemple, l'attribut „Transportunternehmen“ en INTERLIS est formé l'aide de listes (dans FME indiqué par {}). Cette liste contient trois éléments : xtf\_class, TUNummer und TUAbkuerzung.



Cela correspond typiquement à l'implémentation d'une construction INTERLIS « 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;
  
```

Dans ce cas, un seul élément est défini par STRUCTURE (une STRUCTURE pourrait contenir plusieurs éléments).

Cela facilite l'implémentation qui sera possible à l'aide d'un seul transformateur.

A l'aide d'un AttributeManager, il faut :

- Créer trois nouveaux attributs, qui ne se trouvent pas dans les données sources :

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

Pour la valeur (Attribute Value), utiliser la valeur du nom de la STRUCTURE (de manière complète avec Model.Topic.Class).

Les valeurs peuvent être copiées depuis ici :

*HaltestellenOeV\_V1\_2.HaltestellenOeV.TU*

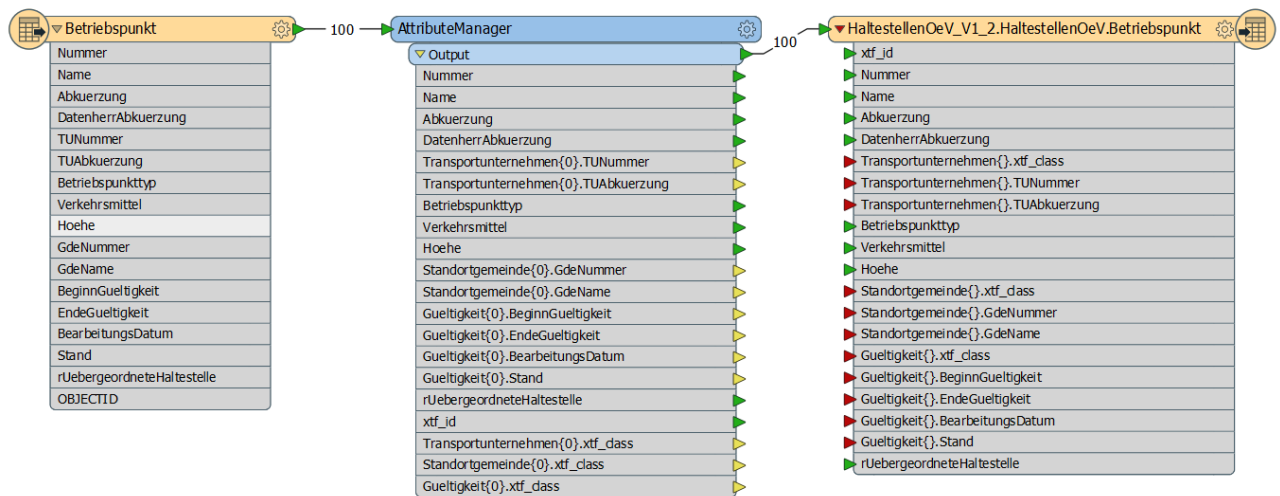
*HaltestellenOeV\_V1\_2.HaltestellenOeV.Gemeinde*

*HaltestellenOeV\_V1\_2.HaltestellenOeV.Gueltigkeit*

- Créer un identifiant unique xtf\_id. Pour cela, reprendre simplement l'OBJECTID.
- Lier les attributs comme ci-dessous :

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

Le workspace ressemble maintenant à cela :



Définir le paramètre «Validate» à No.

Essayer d'exécuter le workspace et regarder le résultat en ouvrant l'XTF avec un éditeur de text.

```

</HEADERSECTION>
<DATASECTION>
  <HaltestellenOeV_V1_2.HaltestellenOeV BID="101">
    <HaltestellenOeV_V1_2.HaltestellenOeV.Betriebspunkt TID="1">
      <Nummer>8572922</Nummer>
      <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>

```

Cette méthode est une implémentation simple des listes FME, qui dans ce cas ne sont constitués que d'un seul élément.

### 3. EX. 3 – LECTURE DONNÉES MGDM ET EXPORT EN GDB

#### 3.1 Introduction

J'ai reçu un fichier de données INTERLIS 2.3 selon le MGDM 190.1 (espace réservé aux eaux). Je souhaite exporter ce fichier dans une GDB pour intégrer les données dans mon projet ArcGIS Pro.

#### 3.2 Buts

- Utilisation de FME pour lire un XTF et l'exporter dans une GDB
- Prise en compte d'attributs de type STRUCTURE

#### 3.3 Données de l'exercice

Le fichier *ID190\_1\_Espace\_reserve\_aux\_eaux\_V1.1.xtf* provient d'un export du canton du Jura, effectué sur geodienste.ch.

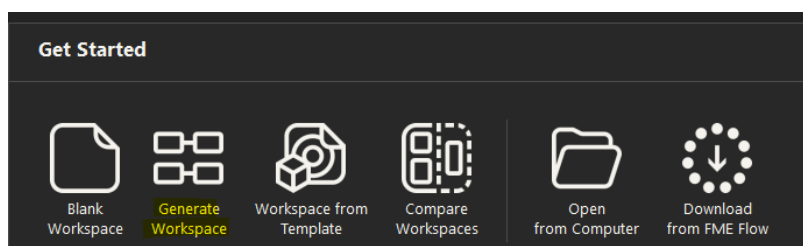
#### 3.4 Marche à suivre

1. Pour commencer, nous allons regarder à quoi ressemble le fichier ILI du modèle correspondant Gewaesserraum\_V1\_1.ili.

Après avoir ouvert le fichier dans un éditeur de texte, répondez aux questions suivantes :

- Combien de CLASS contient le modèle ?
- Quelle CLASS contient une géométrie ? De quel type de géométrie s'agit-il ?
- Quels attributs sont liés à un DOMAIN ?
- Quel est la cardinalité de l'association DokumentGewR ?
- Quels sont les quatre attributs qui sont de type texte multilingue (MultilingualText) ?

2. Ouvrir FME Workbench, puis cliquer sur « Generate Workspace »



3. Pour le Reader, choisir le format INTERLIS et le fichier XTF situé dans le dossier de l'exercice.

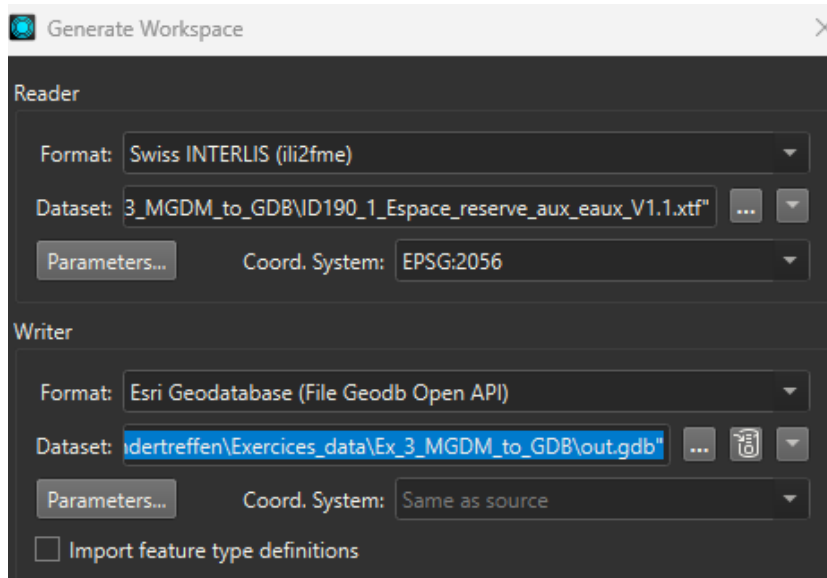
Dans les paramètres du Reader, il faut définir le paramètre Validate à No. Cette fonction permet de vérifier que le fichier XTF en lecture est conforme au modèle ILI correspondant. Dans notre cas le fichier est conforme et il est préférable de désactiver ce paramètre pour des questions de performances.



De plus, il faut définir le système de coordonnées du Reader : LV95, EPSG :2056.

Pour le Writer, choisir le format GDB (Open API) et définir une GDB en sortie « out.gdb » dans le dossier de l'exercice.

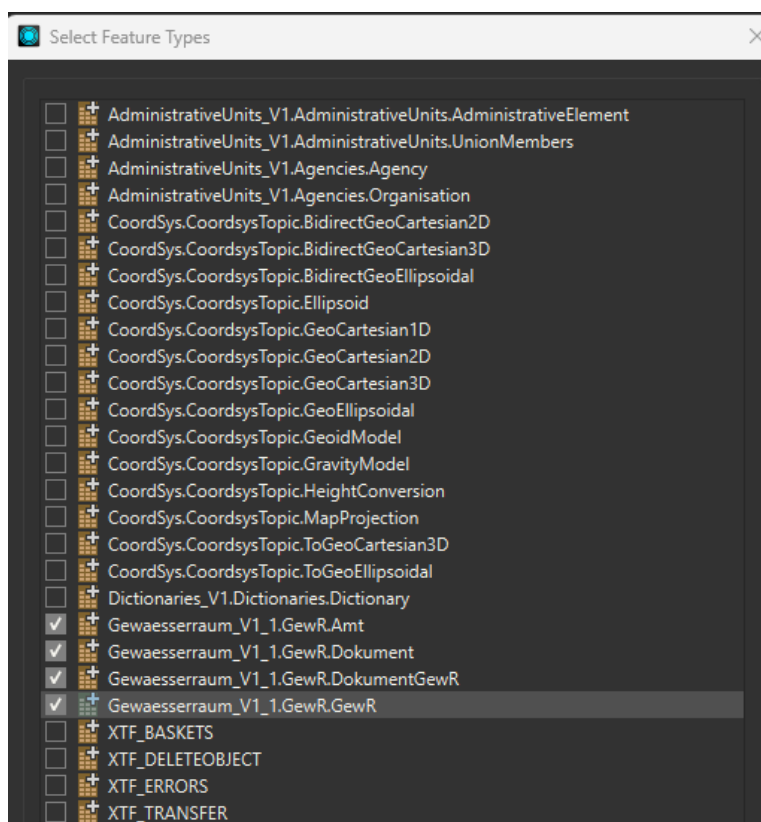
La fenêtre doit ressembler à ça :



#### 4. Cliquer sur OK.

La fenêtre « Select Feature Types » s'ouvre. Elle contient toutes les classes qui se trouvent dans notre modèle « Gewaesserraum\_V1\_1 » ainsi que dans tous les modèles qui sont importés dans ce ILI.

Dans notre cas, on ne s'intéresse qu'aux quatre classes issues de notre modèle « Gewaesserraum\_V1\_1 ».



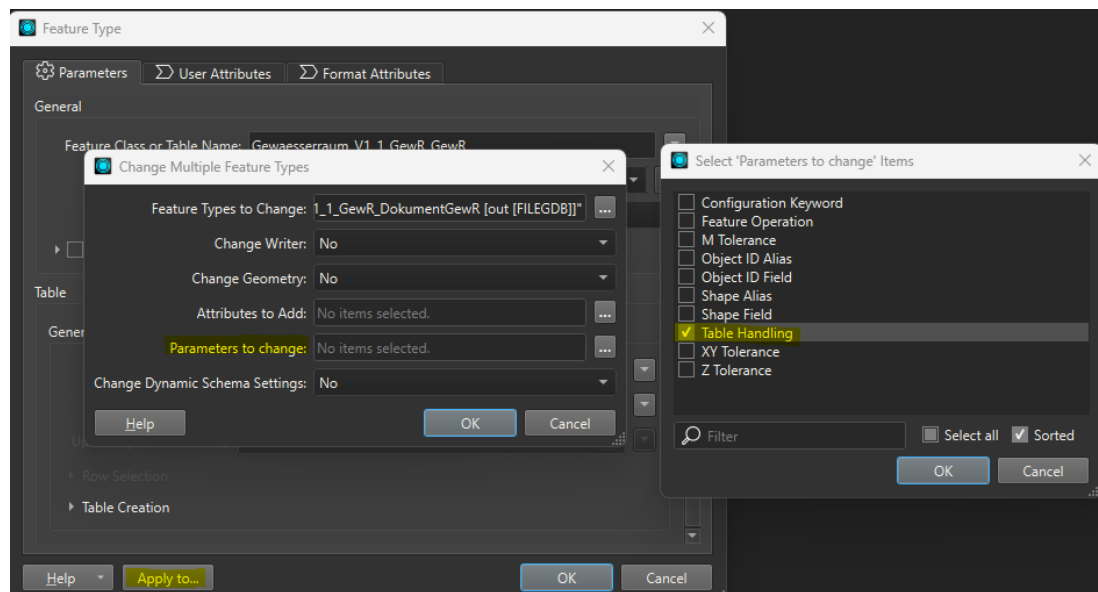
On voit que le plugin ili2fme nous propose également la classe DokumentGewR. Comme on l'a vu précédemment, il s'agit d'une association de cardinalité n : m, qui est donc matérialisée par une table de relations.

5. Nous obtenons la fenêtre suivante :



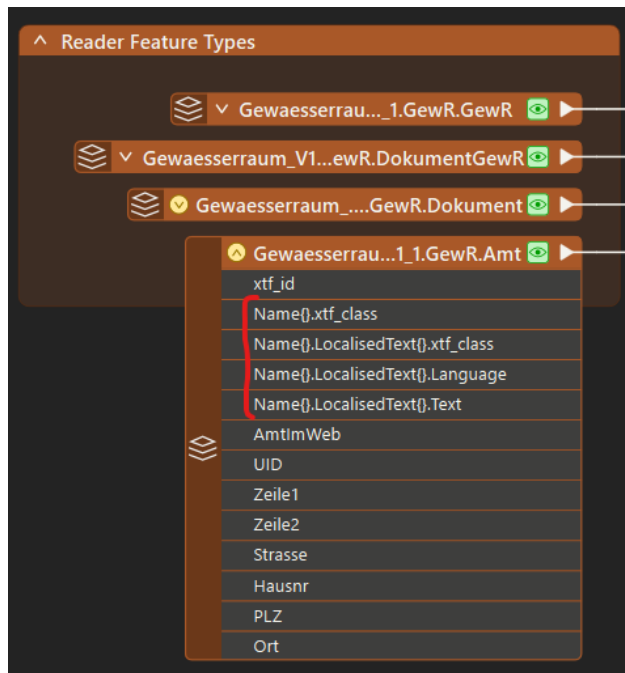
En utilisant le Generate Workspace, l'avantage est que FME crée automatiquement les classes correspondantes dans le Writer et effectue le mapping.

6. Cliquer sur le FeatureType GewR, puis définir « Table Handling » = « Drop and Create ». Pour appliquer ce changement à tous les Feature Types, cliquer sur « Apply to... » en bas de la fenêtre. Dans « Parameters to change », sélectionner « Table Handling ». Puis cliquer trois fois sur OK. Si une confirmation est demandée, cliquer sur Yes.



Cela permettra de récréer les classes dans la GDB à chaque exécution de workspace.  
Exécuter le workspace.

7. Cliquer sur la flèche pour ouvrir la liste des attributs de la classe Amt.



On voit que l'attribut « Name », qui est défini comme texte multilingue dans le ILI, est une liste dans FME. On voit également qu'un deuxième niveau de liste existe : « LocalisedText ».

Cliquer sur l'œil, pour visualiser les données du Feature Type « Amt » :

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)

En regardant de plus près la structure de l'attribut « Name », on voit que la liste « Name » contient un seul objet (ici en jaune). Puis, la liste « LocalisedText » contient deux objets (en vert).

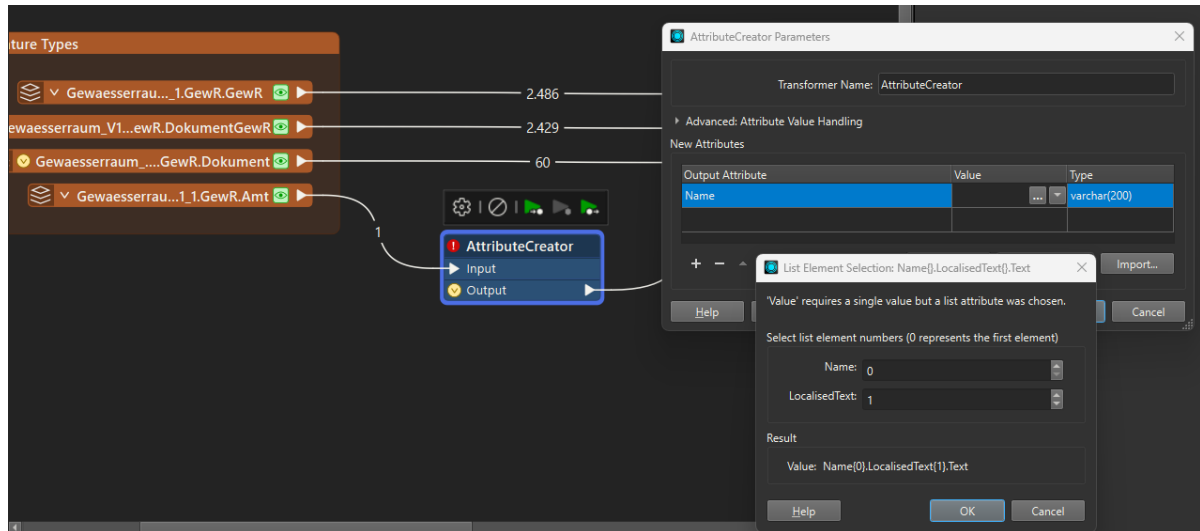
Le premier élément (index 0) est générique. Le deuxième (index 1) contient la valeur en français, qui est ici la même que la valeur générique, soit « Service du développement territorial (SDT) ». Cette structure pourrait permettre, si besoin, d'ajouter des éléments à la liste pour intégrer le nom dans d'autres langues.

- Nous allons maintenant créer un attribut « Name » contenant la valeur en français de cet attribut multilingue.

Ajouter un AttributeCreator après le Reader Feature Type « Amt ».

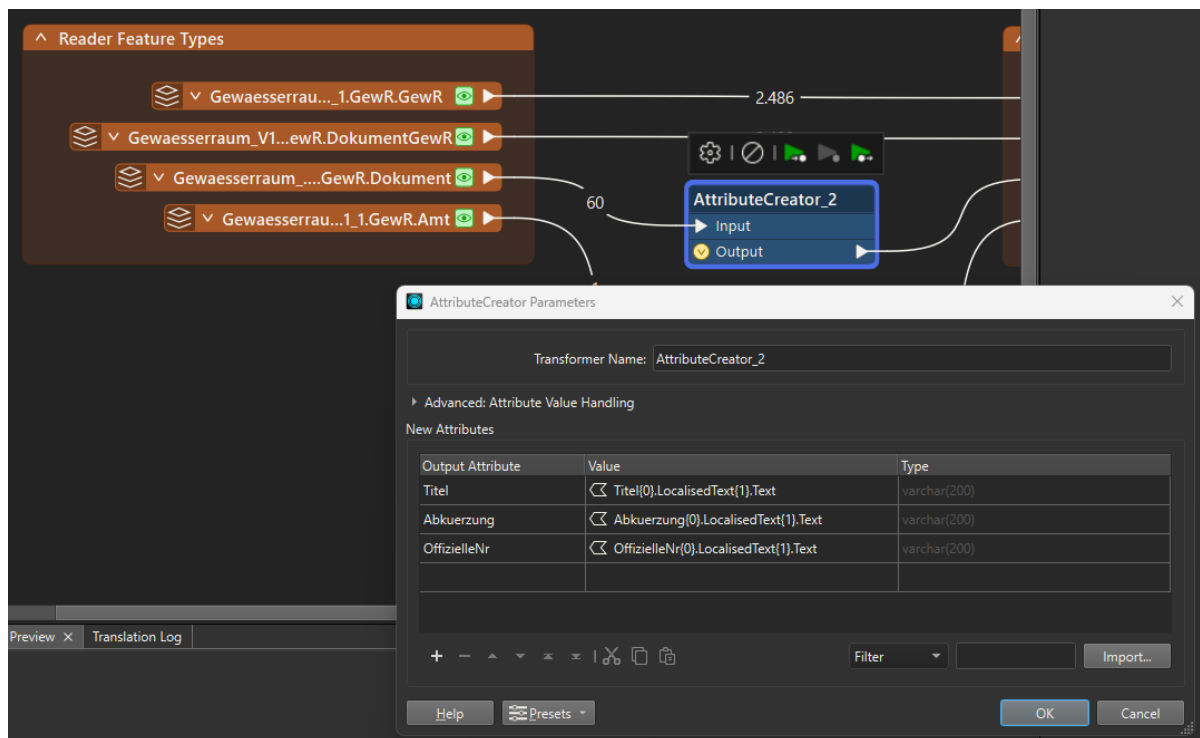
Créer un nouvel attribut « Name ». Comme valeur, sélectionner l'attribut « Name{ }.LocalisedText{ }.Text ».

Comme on veut aller chercher un élément précis dans ces deux listes, il est nécessaire d'indiquer quel index de chaque liste doit être utilisé. Comme vu précédemment, nous définissons 0 et 1 :



Cliquer deux fois sur OK.

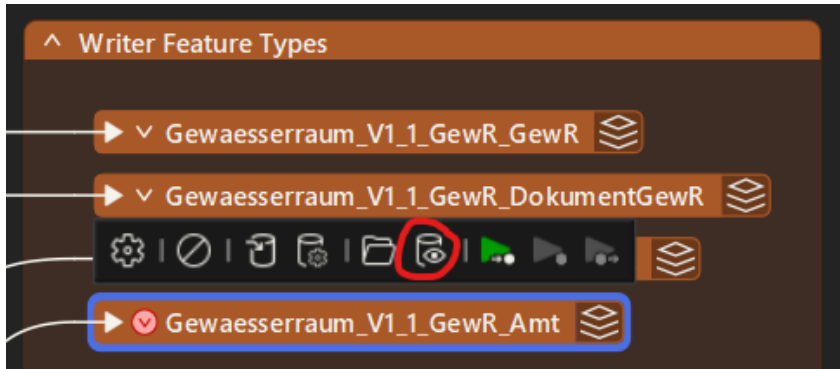
- Effectuer la même démarche pour les trois attributs Titel, Abkuerzung et OffizielleNr de la classe Dokument :





# 10. Exécuter le workspace.

Effectuer un clique-droit sur le Writer Feature Type, puis cliquer sur le bouton ci-dessous (en rouge) pour visualiser les données écrites.



Dans « Amt » et « Dokument », les attributs en question sont visibles, avec une valeur correspondant à la valeur en français de l'attribut multilingue.

Si on consulte la GDB en sortie avec ArcGIS Pro, les quatre classes ont été correctement créées :



## 4. EX. 4 – FACULTATIF - EXPORT MGDM AVEC FME

### 4.1 Introduction

A partir d'une GDB contenant les données du canton de Fribourg, l'objectif de l'exercice est d'exporter ces données vers le MGDM correspondant avec FME.

Le modèle est *SanierungWasserkraft\_LV03\_V1\_2* (ID 192). En français *Assainissement des centrales hydrauliques*.

### 4.2 Buts

- Utilisation de FME pour exporter une GDB en INTERLIS
- Vérifier le résultat avec iliValidator

### 4.3 Marche à suivre

La GDB en entrée contient les tables suivantes, qui contiennent toutes des objets :



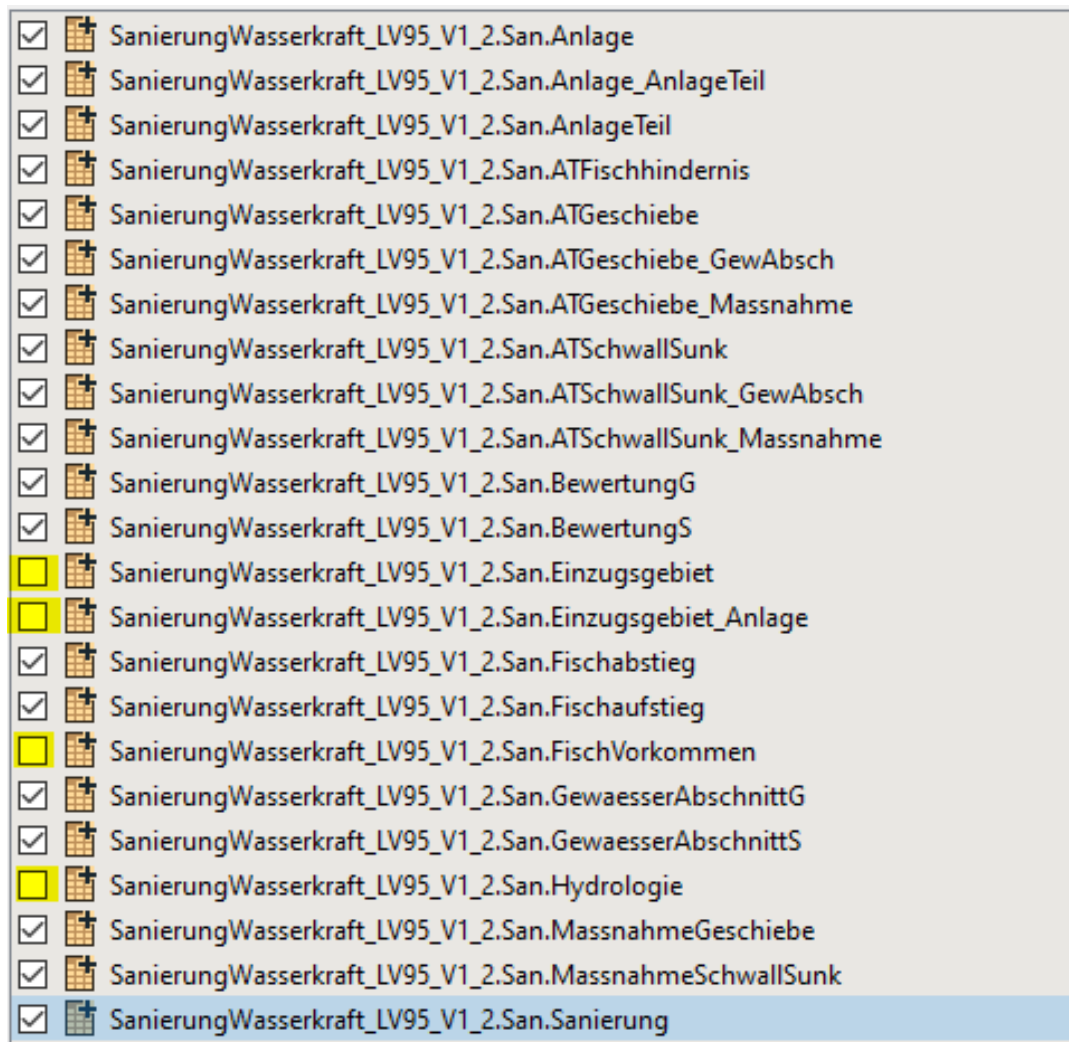
Ces tables sont déjà très proches de la modélisation selon le ILI.

1. Ouvrir un workspace vide
2. Ajouter un reader GDB OpenAPI en chargeant toutes les tables de la GDB. La base de données se trouve dans le dossier GDB\_IN

Avant de rentrer dans le vif du sujet, il est important de « s'imprégner » du modèle INTERLIS :

- Après avoir ouvert le ILI (SanierungWasserkraft\_V1\_2.ili dans le dossier ILI\_OUT) avec Notepad++ :
  - Repérer les différents modèles (MODEL) du fichier
  - Repérer le ou les TOPIC de chaque modèle
  - Repérer les DOMAINS et les STRUCTURE, regarder où ils sont utilisés
  - Repérer les différentes CLASSES et les ASSOCIATIONS existantes entre elles (il y en a beaucoup...)
  - Quelles classes contiennent une géométrie ?
  - A quel type de géométrie correspond la géométrie « StrOrt » ?

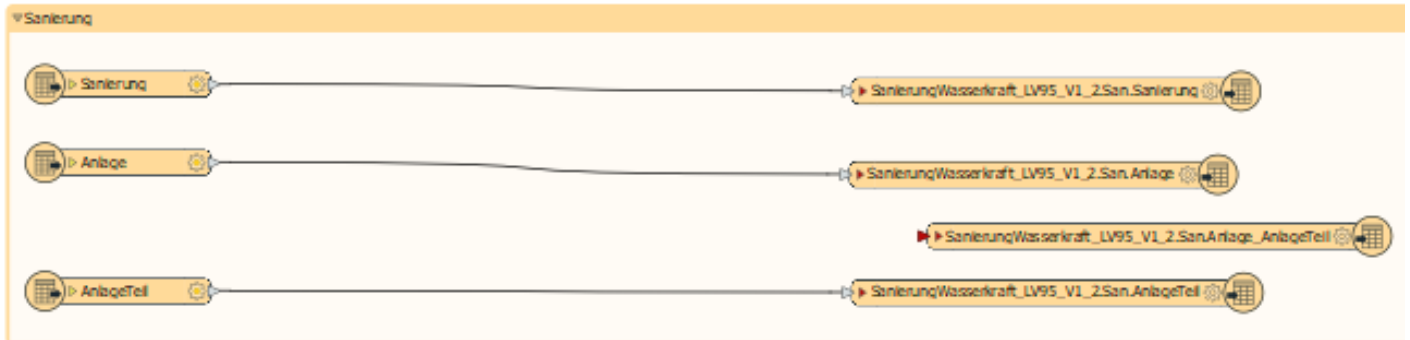
- Ouvrir le PDF « 190206\_Planification\_et\_rapports... .pdf », qui se trouve dans le même dossier.
  - Se rendre à la page 19 et consulter le diagramme UML : essayer de faire le lien entre quelques classes/associations et le modèle ILI.
  - Repérer la zone du diagramme qui concerne :
    - L'assainissement des installations
    - Régime de charriage
    - Eclusées
    - Migration des poissons
  - Se référer aux pages 12-13 du PDF pour comprendre dans les grandes lignes à quoi font référence ces termes.
- 3. Ajouter un writer INTERLIS, en chargeant les classes suivantes du modèle ILI (pas besoin d'importer celles qui ne sont pas dans la GDB) :



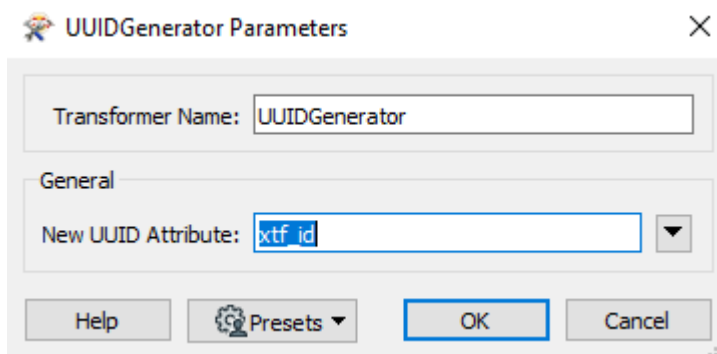
- 4. Répondre aux questions suivantes :
  - Quel est le point commun entre les classes suivantes du modèle INTERLIS:  
Anlage\_AnlageTeil / ATGeschiebe\_GewAbsch / ATGeschiebe\_Massnahme /

ATSchwallSunk\_GewAbsch / ATSchwallSunk\_Massnahme /  
Einzugsgebiet\_Anlage ?

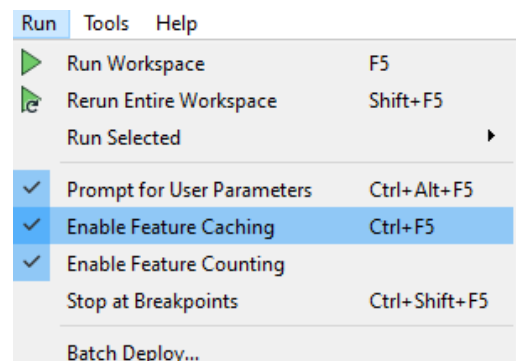
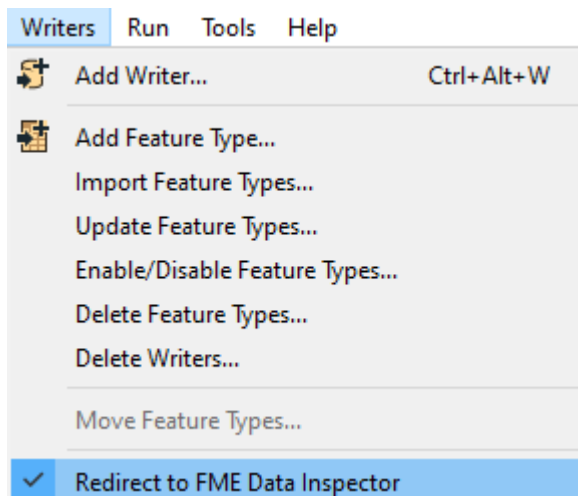
- Pourquoi sont-elles importées dans FME ?
5. Une fois les classes sont importées, nous allons nous concentrer sur la partie « Sanierung » du modèle. Mettre en forme le workspace pour obtenir quelque chose qui ressemble à ça :



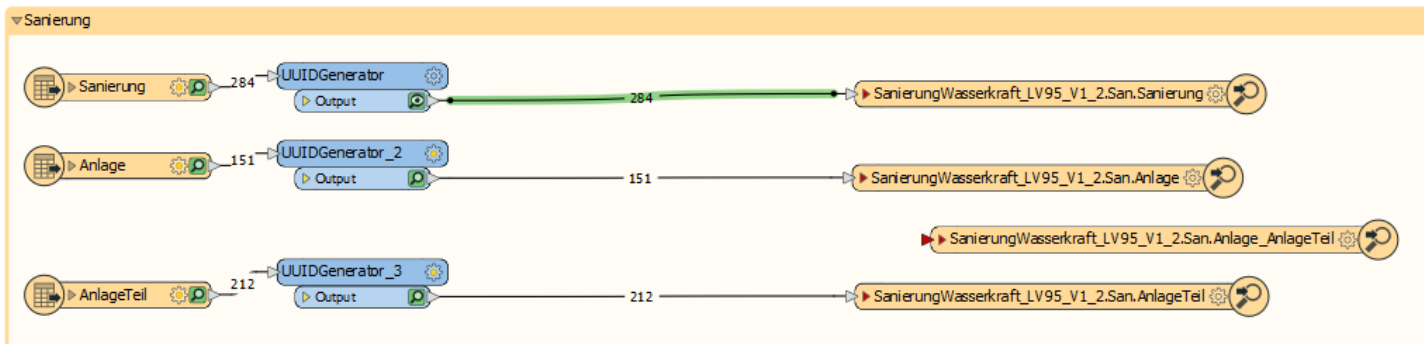
6. Pour chaque classe, il faut déjà générer un identifiant xtf\_id à chaque objet, en ajoutant 3 UUIDGenerator :



7. Dans Writers, activer “Redirect to FME Data Inspector”, ainsi que “Enable Feature Caching” dans Run. Cela est pratique lors du développement.



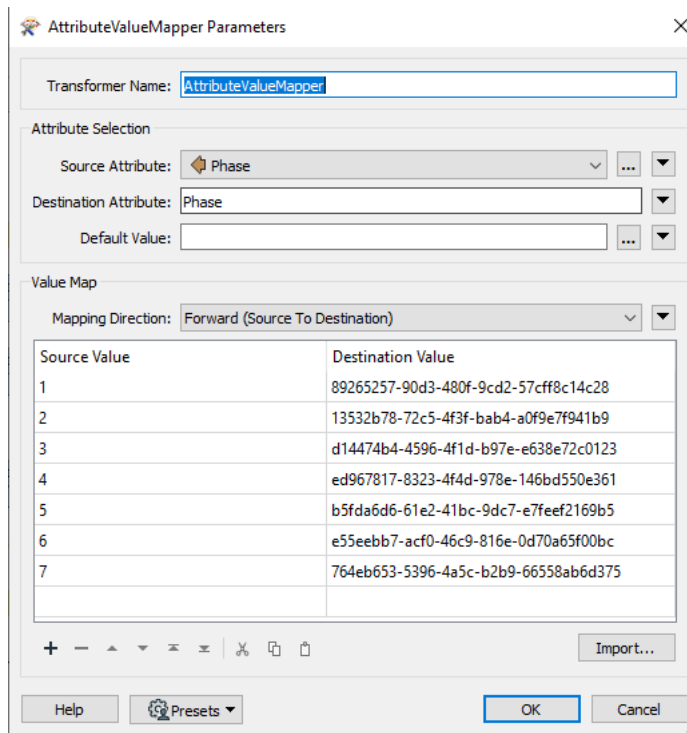
8. Exécuter le workspace, qui ressemble pour l'instant à ça :



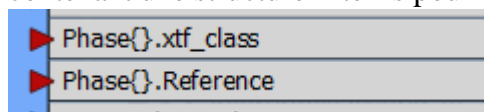
9. Concernant la classe Sanierung, nous voyons dans le ILI que l'attribut Phase est obligatoire et qu'il fait référence au catalogue SanPhaseRef. Cliquer sur la petite loupe à la sortie du premier UUIDGenerator pour visualiser les données à cet endroit. On peut voir que l'attribut Phase a une valeur comprise entre 1 et 7. Si on recherche SanPhase dans le fichier XML des catalogues, on trouve des codes qui ressemblent à ça :

```
<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="":
```

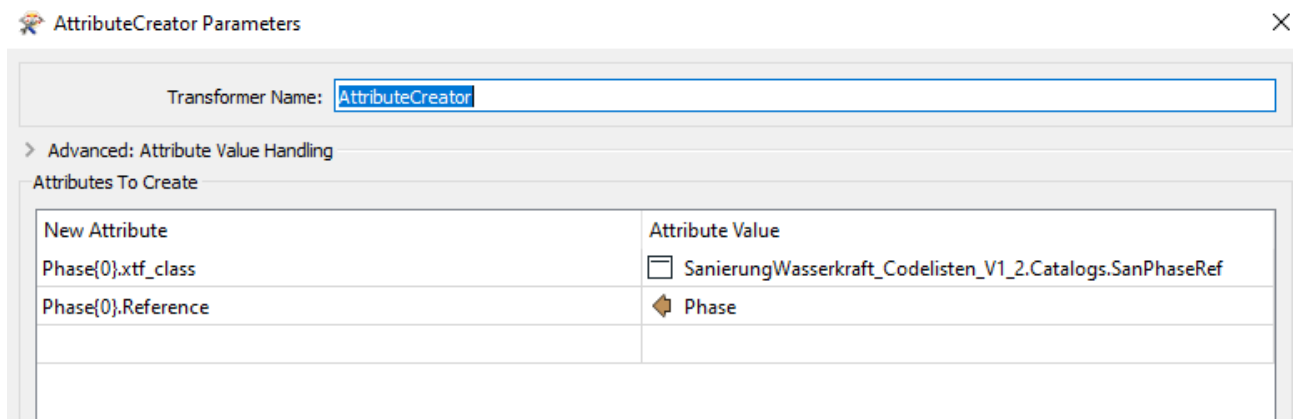
Par exemple ici : « SanPhase7 ». En l'occurrence, c'est le OID qui devra être renseigné dans la valeur attributaire. Nous allons donc utiliser le transformer AttributeValueMapper pour mapper le numéro de phase avec le OID correspondant dans le XML des catalogues. On obtient donc :



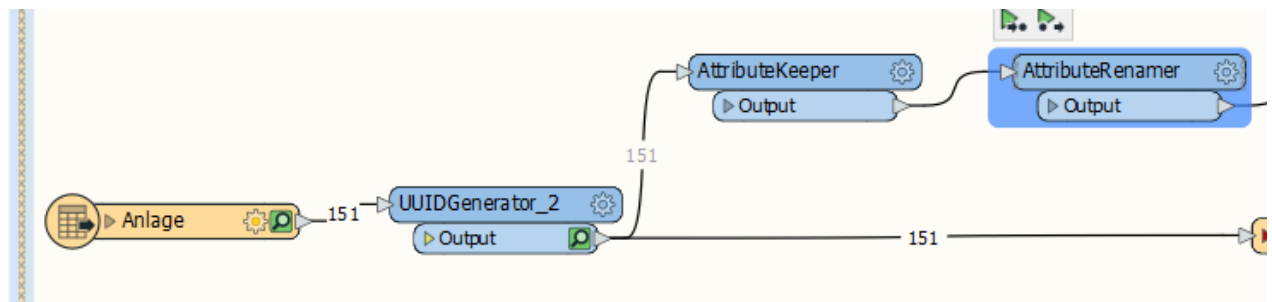
10. Il nous faut maintenant renseigner correctement l'attribut Phase{}, qui est une liste car contenant une structure Interlis pour les catalogues.



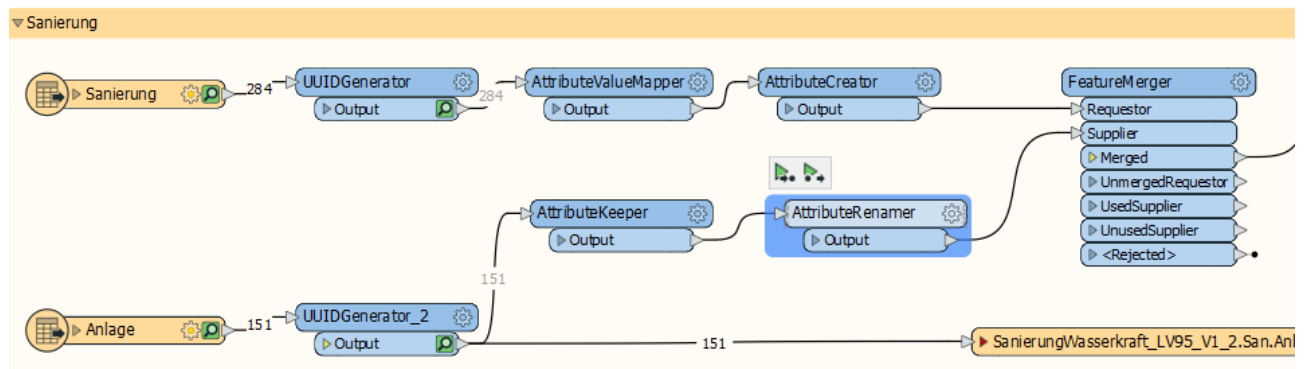
Pour ce faire, on utilise un AttributeCreator, juste après le AttributeValueMapper :



11. Toujours dans la classe Sanierung, on voit que le writer contient l'attribut rBauwerk. En se référant au modèle, on voit qu'il s'agit d'une clé étrangère sur la table Anlage. Dans la GDB en entrée, on voit qu'il s'agit de l'attribut rSanierung, qui contient un lien vers l'attribut AnlageNr de la table Anlage. Comme nous avons besoin d'un identifiant technique comme clé étrangère, nous allons utiliser un FeatureMerger pour récupérer cet attribut. Commencer par placer un AttributeKeeper après le UUIDGenerator\_2 pour ne conserver que les attributs xtf\_id et AnlageNr. Placer ensuite un AttributeRenamer pour renommer xtf\_id en rBauwerk.



Puis insérer le FeatureMerger de la manière suivante :



#### 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:

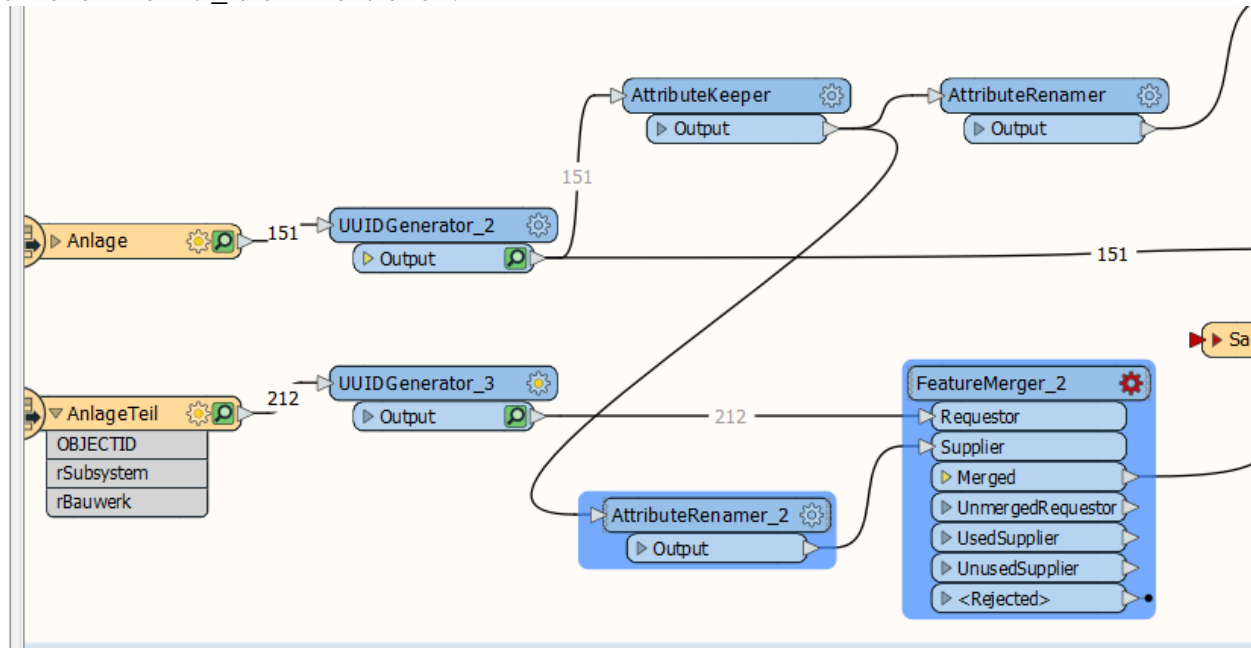
Reject Null and Missing Keys:

☐ Process Duplicate Suppliers

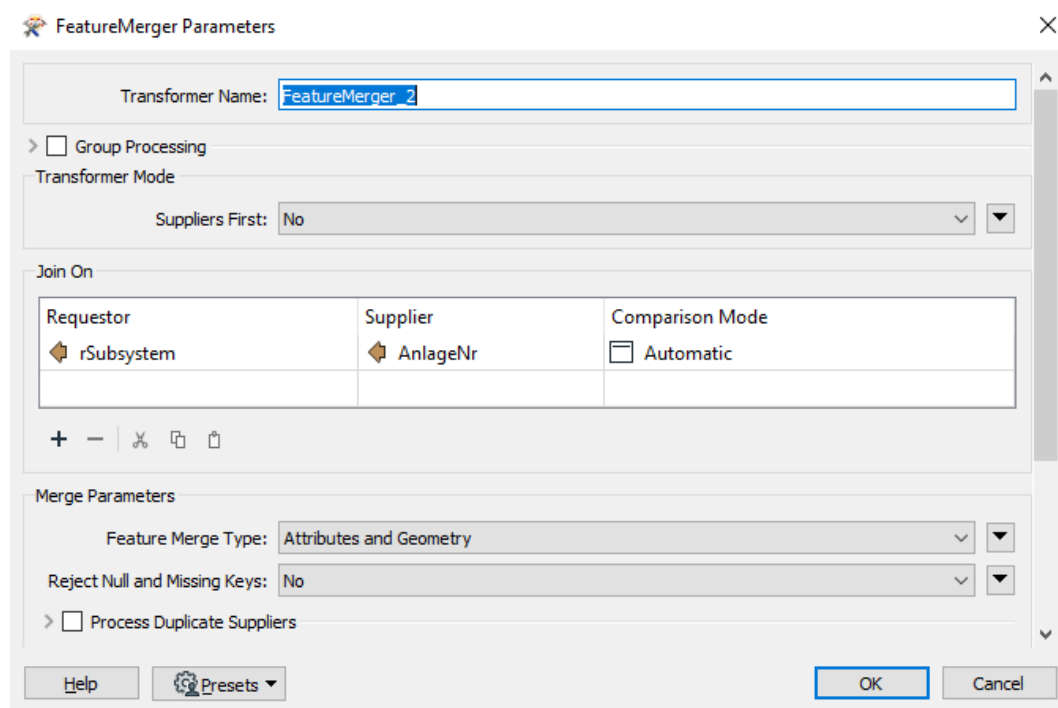
De cette manière, l'attribut rBauwerk est maintenant correctement renseigné.

12. Une opération similaire est à réaliser pour l'attribut Type de la classe Anlage, qui est aussi un attribut obligatoire. Cette étape est facultative.
13. Il nous reste à remplir la table AnlageTeil et la table de relations Anlage\_AnlageTeil. Dans la GDB, on voit que la table AnlageTeil contient deux attributs :
- rSubsystem, qui est une clé étrangère sur AnlageNr
  - rBauwerk, qui est l'identifiant métier de AnlageTeil

Comme précédemment, il faut aller rechercher la clé technique de AnlageTeil, en utilisant un FeatureMerger de la manière suivante : commencer par ajouter un AttributeRenamer pour renommer xtf\_id en rKontroller :

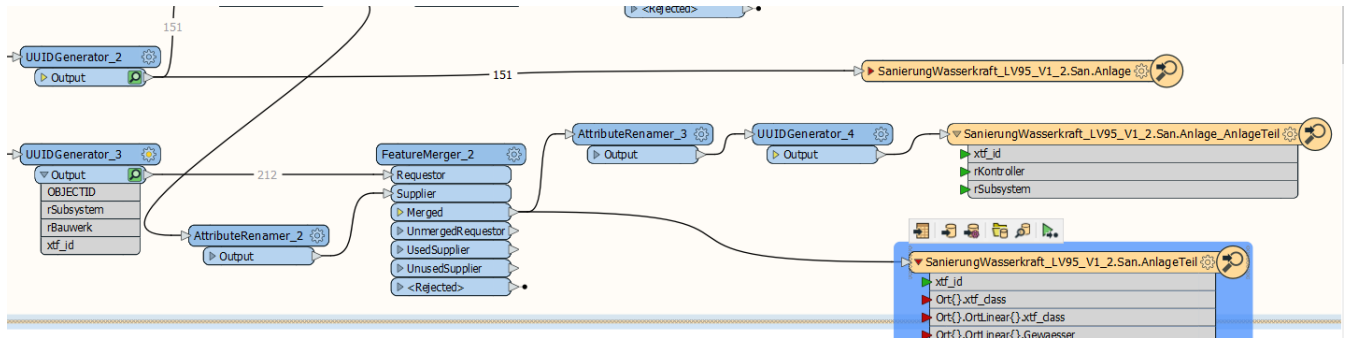


Puis ajouter le FeatureMerger. Ne pas oublier de définir « Attributes and Geometry » pour le Feature Merge Type. La géométrie récupérée de Anlage sera utilisée plus tard.

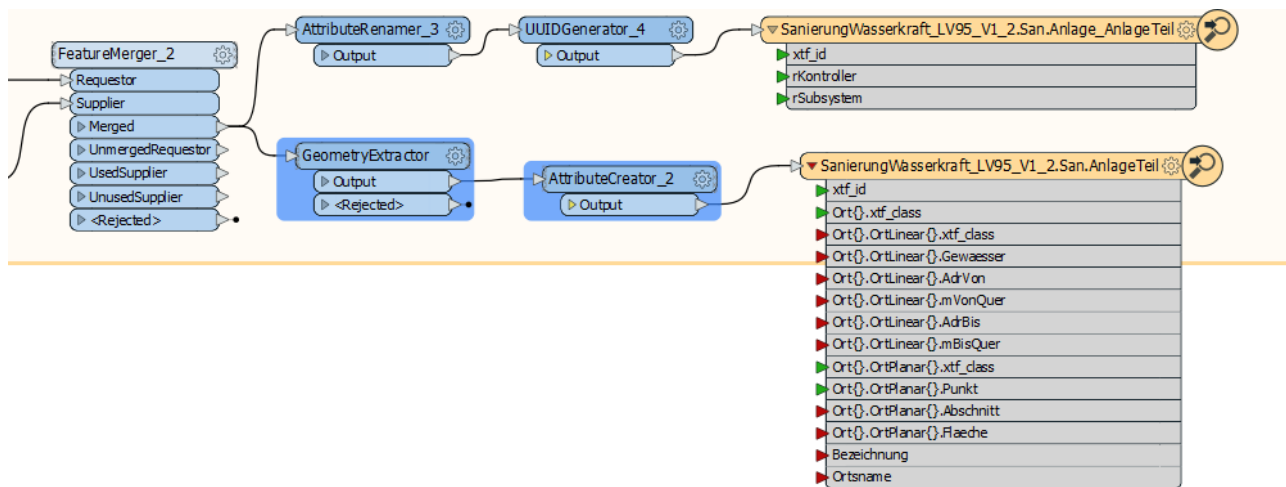




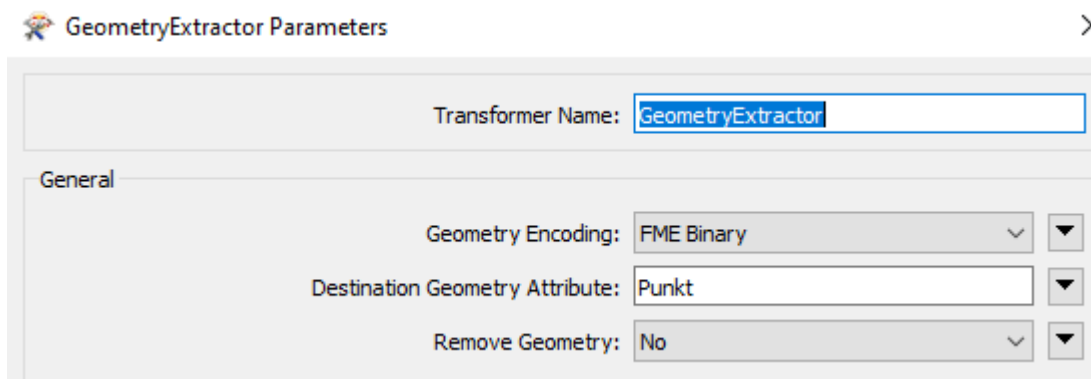
14. Pour renseigner la table de relations Anlage\_AnlageTeil, il nous faut un AttributeRenamer à la sortie du FeatureMerger pour renommer rBauwerk en rSubsystem. Il faut encore ajouter un UUIDGenerator pour abtenir un xtf\_id pour chaque objet.

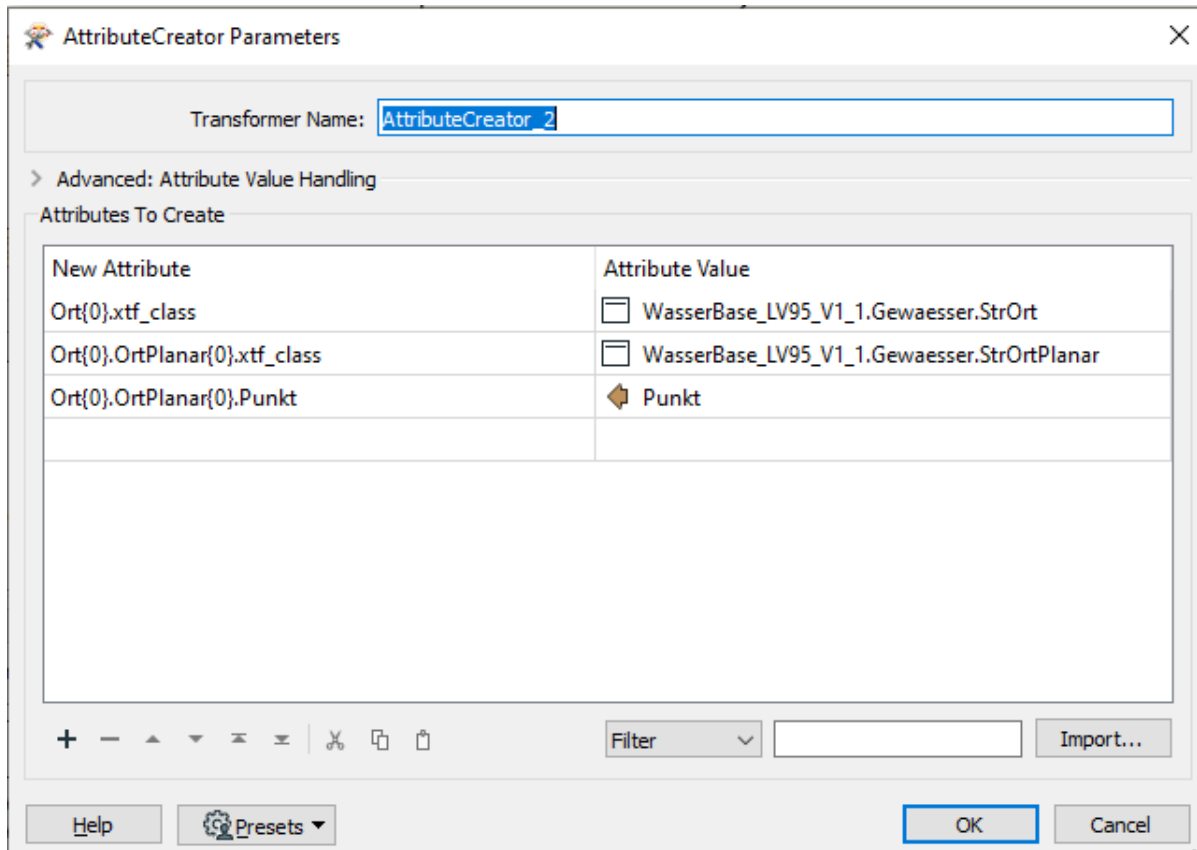


15. La classe Interlis AnlageTeil contient une géométrie (attribut Ort{}), mais qui n'est pas obligatoire. Pour l'exercice nous allons tout de même utiliser la géométrie point que l'on a récupérer avec le FeatureMerger pour renseigner une géométrie. Commencer par placer un GeometryExtractor, puis un AttributeCreator...



...avec les paramètres suivants :





16. Dans les paramètres du writer, définir le fichier xtf à écrire. Par exemple : /ILI\_OUT/SanierungWasserkraft\_V1\_2.xtf

Désactiver le Validate dans le writer Interlis.

Enlever le Redirect to Visualizer et tenter de faire tourner le workspace.

Vérifier que le xtf est créé et que la structure semble correcte.

17. Le résultat peut ensuite être passé au checker iliValidator. Une version est disponible en ligne si besoin (ilicop.ch). A ce stade, un grand nombre d'erreurs existent au checker. On entre donc dans la phase de correction du fichier XTF jusqu'à ce qu'il ne donne plus d'erreur au checker...

Essayer de comprendre les erreurs dans le log et de les résoudre si possible.