

ATELIER SION - 22 OCTOBRE 2026

MODEL BAKER



Commençons ...

QUE FAISONS-NOUS AUJOURD'HUI ?

ISABEL KIEFER ET
DAMIANO LOMBARDI

QUI ÊTES-VOUS ?

- D'où viens-je ?
- Quel est mon travail principal ?
- Je suis **Model Baker** ... :
- A. ... **Zéro**: Qu'est-ce que Model Baker ?
- B. ... **Utilisateur**: Ai-je déjà créé des projets ?
- C. ... **Héros**: Ai-je déjà travaillé avec des conteneurs / validation, etc. ?

OBJECTIFS DE L'ATELIER

- Représenter des modèles INTERLIS dans la base de données et importer / exporter / valider des données
- Être capable de travailler avec des conteneurs et des OID dans QGIS
- Savoir ce que sont les méta-attributs et les stratégies d'héritage, etc.

PROGRAMME (1)

Temps	Contenu
09.00	Introduction et environnement 🧑🏫
10.00	Importation de modèles 👥
10.30	Méta-attributs 🧑🏫💻
11.00	Pause
11.15	Héritages et cartographie 🧑🏫💻
12.00	Dîner

PROGRAMME (2)

Temps	Contenu
13.00	Importation et exportation de données 👥 Conteneurs et ensembles de données 🧑🏫💻
13.45	Saisie et validation 👥💻
15.15	Pause
15.30	Usability Hub 🧑🏫👥💻
16.30	modèles de traduction 🧑🏫
16.45	Conclusion, questions et suggestions

Courte introduction...

QU'EST-CE QUE MODEL BAKER ?

UN GÉNÉRATEUR DE PROJETS QGIS

Création rapide d'un **projet QGIS** à partir d'un modèle de données physique.

Analyse la structure existante et configure un projet QGIS avec toutes les informations disponibles.

UN GÉNÉRATEUR DE PROJETS QGIS OPTIMISÉ POUR *INTERLIS*

Les modèles définis dans INTERLIS fournissent des **méta-informations supplémentaires** telles que les domaines, les unités d'attributs ou les définitions orientées objet des tables.

Celles-ci peuvent être utilisées pour optimiser davantage la configuration du projet.

UNE STATION DE CONTRÔLE ILI2DB

```
java -jar /home/dave/dev/opengisch/QgisModelB
--dbusr postgres --dbpwd ***** --dbdatabase
--createFk --createFkIdx --coalesceMultiSurfa
--expandMultilingual --createTypeConstraint -
--models Wildruhezonen_LV95_V2_1
```

ET C'EST UNE BIBLIOTHÈQUE

Peut être utilisé comme un cadre. Comme le fait [Asistente LADM-COL](#).

Ou TEKSI [Script TEKSI wastewater](#)

... et est également disponible en tant que package Python :

```
pip install modelbaker
```

OÙ TROUVER QUOI ?

DÉPÔTS DE CODE

Plugin QGIS

<https://github.com/opengisch/QgisModelBaker>

Dépôt de code Bibliothèque Backend

<https://github.com/opengisch/QgisModelBakerLibrary>

REPORT DE BUGS ET DEMANDES DE FONCTIONNALITÉS

<https://github.com/opengisch/QgisModelBaker/issues>

- Vérifier
- Saisir
 - Description du problème
 - Infos supplémentaires
- Attendre ou financer 😊

DOCUMENTATION



modelbaker.ch

opengisch.github.io/QgisModelBaker

De quoi avons-nous besoin...

ENVIRONNEMENT

QGIS ET BASES DE DONNÉES

- QGIS 3.40 ou supérieur
- PostgreSQL + PostGIS
- Geopackage possible

INTERNET ?

A compléter

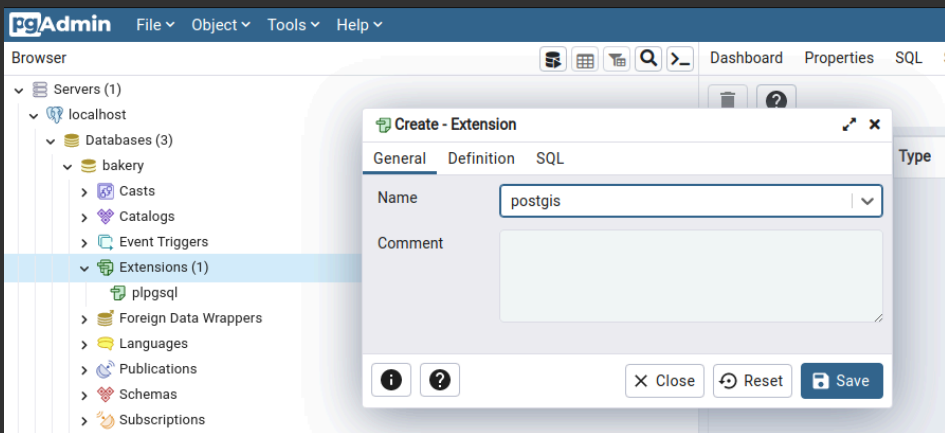
Nom :

Code :

Pad : <https://pad.opengis.ch>

EXTENSIONS AVEC PGADMIN

- Ajouter une extension :
- `postgis`
- `uuid-osp`



EXTENSIONS AVEC LE CLIENT PSQL

```
$ psql -Upostgres -hlocalhost -dbakery
postgres=# CREATE DATABASE databasename;
postgres=# \connect databasename;
databasename=# CREATE EXTENSION postgis;
databasename=# CREATE EXTENSION uuid-oss;
```

Si ce n'est pas déjà installé...

INSTALLATION DE POSTGIS

Windows

<https://postgis.net/install/>

Là, selon la version de PostgreSQL, trouver `postgis-bundle-pg14x64-setup-3.3.2-2.exe`, le télécharger et l'installer.

Mac

Utiliser de préférence **posgres.app** (PostGIS y est déjà inclus)
<https://postgresapp.com/de/downloads.html>

Linux

```
sudo apt install postgis
```

OPENGIS.CH POSTGRESQL

Hôte	modelbaker-workshop-exoscale-d9c1bf3f-51a3-4ab8-afac-6cb37d8a8f.a.aivencloud.com
Port	21699
Utilisateur	testbaker
Mot de passe	superbaker
BD	gis

(actif uniquement aujourd'hui)

Utilisez toujours votre abréviation de nom comme préfixe pour les schémas :

```
sigdav_ (pour David Signer)
```

JAVA

Avez-vous Java installé?

```
java --version
```

... il y a une réponse, alors c'est bien 😊

Téléchargement pour Windows / Mac

Ici <https://adoptium.net/de/>

Ou sous Linux (openJDK)

```
sudo apt-get install openjdk-18-jre
```

ou

```
sudo apt install install default-jre
```

TÉLÉCHARGEMENT DES DONNÉES DE DÉMONSTRATION

DONNÉES UTILISATEUR

[HTTPS://GEODIENSTE.CH/?LOCALE=FR](https://geodienste.ch/?LOCALE=FR)

- Kataster der belasteten Standorte (Cadastre des sites pollués)
- Spécification
- Version 1.5
- AtomFeed + Service de téléchargement OpenSearch
- Téléchargement de **SH** et **TG**

QGIS MODEL BAKER

- Installer Model Baker
- Installer Swiss Locator

PRÊT À DÉMARRER 🚀

Commençons...

INTERMÈDE

... laissons-nous divaguer

INTERMÈDE 1 : MÉTA-ATTRIBUTS

QUE SONT LES MÉTA-ATTRIBUTS ?

Les méta-attributs permettent de compléter les modèles INTERLIS avec des **informations supplémentaires** qui ne sont **pas prévues** dans la spécification actuelle d'INTERLIS.

Voir [*eCH-0117 Méta-attributs pour les modèles INTERLIS*](#)

EXEMPLE AVEC MODEL BAKER (1)

Expression d’affichage spécifique pour la table de la classe ZustaendigkeitsKataster.

Relation Reference ▾

Display expression ⓘ 'Nr: '|| uid ||' / '|| katastername ▾ ⓧ

Relation belasteter_standort_zustaendigkeitkataster_fkey (ZustaendigkeitKataster) ▾

MÉTA-ATTRIBUTS VS COMMENTAIRE

Un commentaire commence par **!!** et se termine par une fin de ligne.

Un méta-attribut commence également par **!!** mais suivi de **@**

SYNTAXE DANS LE MODÈLE INTERLIS

```
!!@<name>=<value>
```

Suivi de l’élément référencé (MODEL, TOPIC, CLASS, etc.)

EXEMPLE AVEC MODEL BAKER (2)

Méta-attribut spécifique de Model Baker :

```
!!@qgis.modelbaker.dispExpression='<QGIS Expr
```

EXEMPLE AVEC MODEL BAKER (3)

Définition dans le modèle **KbS_V1_5**

```
1 MODEL KbS_V1_5 (de) =
2
3   TOPIC Belastete_Standorte =
4
5     !!@qgis.modelbaker.dispExpression = "'Nr:
6     CLASS ZustaendigkeitKataster =
7     [...]
```

Relation Reference ▾

Display expression ⓘ 'Nr: '|| uid ||' / '|| katastername ▾ ⓧ

Relation belasteter_standort_zustaendigkeitkataster_fkey (ZustaendigkeitKatas ▾

EXEMPLE AVEC MODEL BAKER (3)

Définition dans le modèle **KbS_V1_5**

```
1 MODEL KbS_V1_5 (de) =
2
3   TOPIC Belastete_Standorte =
4
5     !!@qgis.modelbaker.dispExpression = "'Nr:
6     CLASS ZustaendigkeitKataster =
7     [...]
```

Relation Reference ▾

Display expression ⓘ 'Nr: '|| uid ||' / '|| katastername ▾ ⓧ

Relation belasteter_standort_zustaendigkeitkataster_fkey (ZustaendigkeitKatas ▾

LISTES DE MÉTA-ATTRIBUTS

- [Compilateur INTERLIS \(ili2c\)](#)
- [ili2db](#)
- [Valideur INTERLIS \(ilivalicator\)](#)
- [Gestionnaire INTERLIS \(ilimanager\)](#)
- [QGIS Model Baker](#)

(incomplet) Voir aussi <https://interlis.discourse.group/t/uebersicht-interlis-meta-attribute/70>

LES MÉTA-ATTRIBUTS DANS LE MODÈLE SONT FORMIDABLES

Mais nous ne pouvons pas simplement bricoler le modèle !

FICHER DE MÉTA-ATTRIBUTS SUPPLÉMENTAIRE

... connu comme “fichier de configuration”, “fichier de méta-attributs”, “fichier INI”, “fichier TOML”...

*En réalité au format INI, mais généralement enregistré sous **.toml**.*

À ne pas confondre avec le *fichier de méta-configuration* pour “contrôler” ili2db (également en relation avec UsablLty Hub).

Exemple avec Model Baker

Définition dans le fichier de configuration pour **KbS_V1_5**

```
[ "KbS_V1_5.Belastete_Standorte.Zustaendigkeit"
qgis.modelbaker.displayExpression = "'Nr: ' || ui
```

Relation Reference

Display expression ⓘ 'Nr: ' || uid || ' / ' || katastername

Relation belasteter_standort_zustaendigkeitkataster_fkey (ZustaendigkeitKataster_fkey)

CONTRÔLER LES VALIDATIONS AVEC DES MÉTA-ATTRIBUTS

Juste pour information...

```
[ "KbS_V1_5.Standorte.Belasteter_Standort.Nr" ]
# disable mandatory constraint of Katasternum
multiplicity="off"

[ "KbS_V1_5.Standorte.Belasteter_Standort.Cons
msg = "Mind. eine Untersuchungsmassnahme erfa
```

DE QUOI AVONS-NOUS BESOIN POUR KBS_V1_5 ?

- Multi-géométrie construite à partir de **STRUCTURE MultiPolygon**
- “Multilinguisme aplati” sur **STRUCTURE MultilingualUri**

L'attribut de mapping

`ili2db.mapping`

- Sur la classe ou la structure

Les structures sont mappées avec des colonnes au lieu de relations (par exemple, la multi-géométrie).

- Sur l'attribut

Les attributs de structure (`BAG OF`/`LIST OF`) sont mappés avec un type de données au lieu de relations.

Définition dans le modèle INTERLIS

```
1  !!@ili2db.mapping=MultiSurface
2  STRUCTURE MultiPolygon =
3    Polygones: BAG {1..*} OF PolygonStructure
4  END MultiPolygon;
5
6  !!@ili2db.mapping=Multilingual
7  STRUCTURE MultilingualUri =
8    LocalisedText : BAG {1..*} OF LocalisedUr
9    UNIQUE (LOCAL) LocalisedText: Language;
10  END MultilingualUri;
```

Définition dans le modèle INTERLIS

```
1  !!@ili2db.mapping=MultiSurface
2  STRUCTURE MultiPolygon =
3    Polygones: BAG {1..*} OF PolygonStructure
4  END MultiPolygon;
5
6  !!@ili2db.mapping=Multilingual
7  STRUCTURE MultilingualUri =
8    LocalisedText : BAG {1..*} OF LocalisedUr
9    UNIQUE (LOCAL) LocalisedText: Language;
10  END MultilingualUri;
```

Définition dans le fichier de méta-attributs supplémentaire

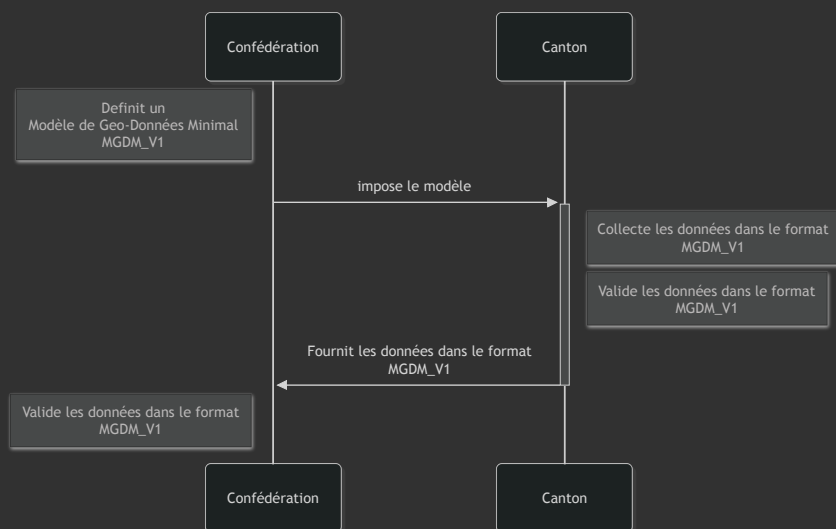
```
[KbS_V1_5.Belastete_Standorte.MultiPolygon]
ili2db.mapping = "MultiSurface"

[KbS_V1_5.Belastete_Standorte.MultilingualUri]
ili2db.mapping = "Multilingual"
```

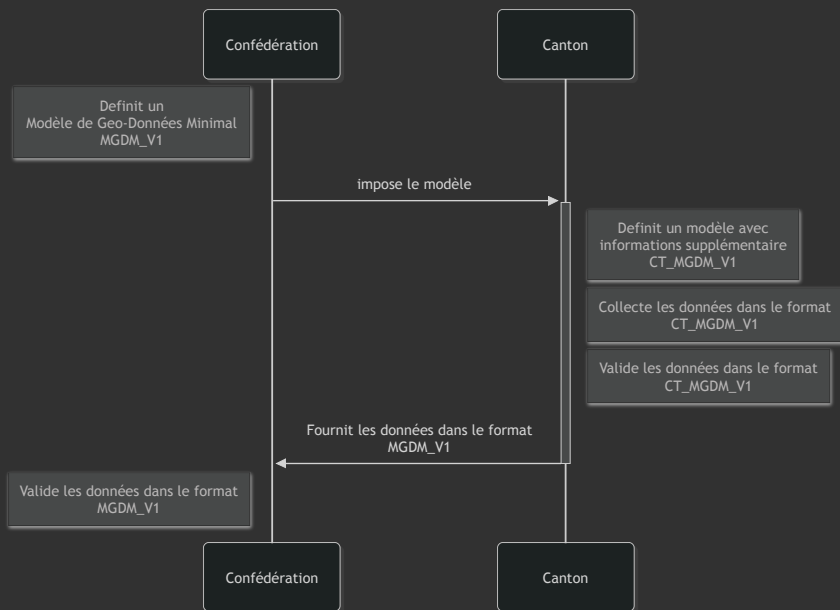
PRÊT POUR L'EXERCICE 1 🚀

INTERMÈDE 2.1 : HÉRITAGES DANS INTERLIS

CAS D'UTILISATION : TRAVAILLER AVEC KBS_V1_5



CAS D'UTILISATION : TRAVAILLER AVEC UNE EXTENSION



EXEMPLE PLUS SIMPLE Gebaeudeinventar_V1

```
1  MODEL Gebaeudeinventar_V1 (de)
2  AT "mailto:signedav@localhost"
3  VERSION "2023-01-19" =
4    IMPORTS GeometryCHLV95_V1;
5
6  TOPIC Gebaeude =
7
8    CLASS Gebaeude =
9      EGID : MANDATORY TEXT*16;
10     Koordinaten : MANDATORY GeometryCHLV95_
11   END Gebaeude;
12
13  CLASS Adresse =
```

EXEMPLE PLUS SIMPLE Gebaeudeinventar_V1

```
1  MODEL Gebaeudeinventar_V1 (de)
2  AT "mailto:signedav@localhost"
3  VERSION "2023-01-19" =
4    IMPORTS GeometryCHLV95_V1;
5
6  TOPIC Gebaeude =
7
8    CLASS Gebaeude =
9      EGID : MANDATORY TEXT*16;
10     Koordinaten : MANDATORY GeometryCHLV95_
11   END Gebaeude;
12
13  CLASS Adresse =
```

EXEMPLE PLUS SIMPLE Gebaeudeinventar_V1

```
1  MODEL Gebaeudeinventar_V1 (de)
2  AT "mailto:signedav@localhost"
3  VERSION "2023-01-19" =
4    IMPORTS GeometryCHLV95_V1;
5
6  TOPIC Gebaeude =
7
8    CLASS Gebaeude =
9      EGID : MANDATORY TEXT*16;
10     Koordinaten : MANDATORY GeometryCHLV95_
11   END Gebaeude;
12
13  CLASS Adresse =
```

EXEMPLE PLUS SIMPLE Gebaeudeinventar_V1

```
3  VERSION 2023-01-19 =
4  IMPORTS GeometryCHLV95_V1;
5
6  TOPIC Gebaeude =
7
8  CLASS Gebaeude =
9    EGID : MANDATORY TEXT*16;
10   Koordinaten : MANDATORY GeometryCHLV95_
11 END Gebaeude;
12
13 CLASS Adresse =
14   Strasse : TEXT*200;
15   Ort : MANDATORY TEXT*200;
16 END Adresse;
```

EXEMPLE PLUS SIMPLE Gebaeudeinventar_V1

```
11 END Gebaeude;
12
13 CLASS Adresse =
14   Strasse : TEXT*200;
15   Ort : MANDATORY TEXT*200;
16 END Adresse;
17
18 ASSOCIATION AdresseGebaeude =
19   GebaeudeAdresse -- {1..*} Adresse;
20   Gebaeude -- {1} Gebaeude;
21 END AdresseGebaeude;
22
23 END Gebaeude;
```

EXEMPLE PLUS SIMPLE Gebaeudeinventar_V1

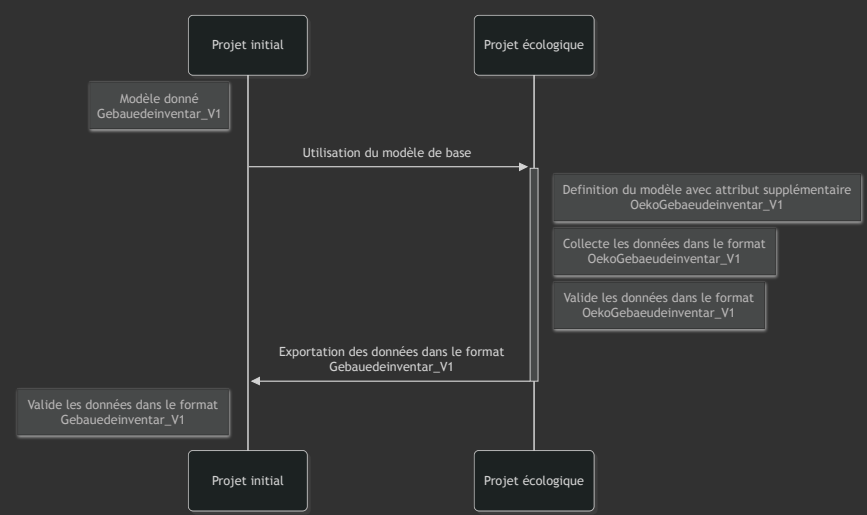
```
3  VERSION 2023-01-19 =
4  IMPORTS GeometryCHLV95_V1;
5
6  TOPIC Gebaeude =
7
8  CLASS Gebaeude =
9    EGID : MANDATORY TEXT*16;
10   Koordinaten : MANDATORY GeometryCHLV95_
11 END Gebaeude;
12
13 CLASS Adresse =
14   Strasse : TEXT*200;
15   Ort : MANDATORY TEXT*200;
16 END Adresse;
```

NOUS AVONS BESOIN DE “PLUS”

Nous devons utiliser un attribut supplémentaire pour un projet écologique : EntsprichtStandard

Mais nous souhaitons toujours renvoyer les données sous forme de Gebaeudeinventar_V1.

CAS D'UTILISATION : PROJET ÉCOLOGIQUE



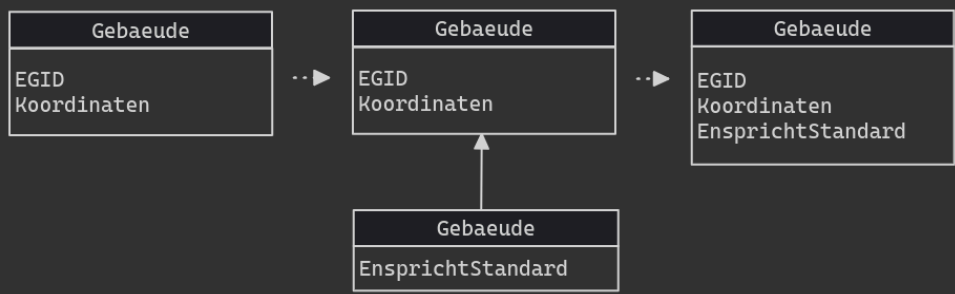
COMMENT “HÉRITONS-NOUS” ?

Étendre une classe (classe de base) en une dérivation “améliorée”.

Une classe étendue (dérivation) a **toutes les propriétés** de la classe de base + “**plus**”.

“**plus**” signifie des **attributs** ou des **restrictions** supplémentaires.

DE Gebaeude DEVIENT UN MEILLEUR Gebaeude



La dérivation en INTERLIS (1)

... avec EXTENDED

```
1 CLASS Gebaeude =
2   EGID : MANDATORY TEXT*16;
3   Koordinaten : MANDATORY GeometryCHLV95_V1.
4 END Gebaeude;
```

```
1 CLASS Gebaeude (EXTENDED) =
2   EntsprichtStandard: BOOLEAN;
3 END Gebaeude;
```

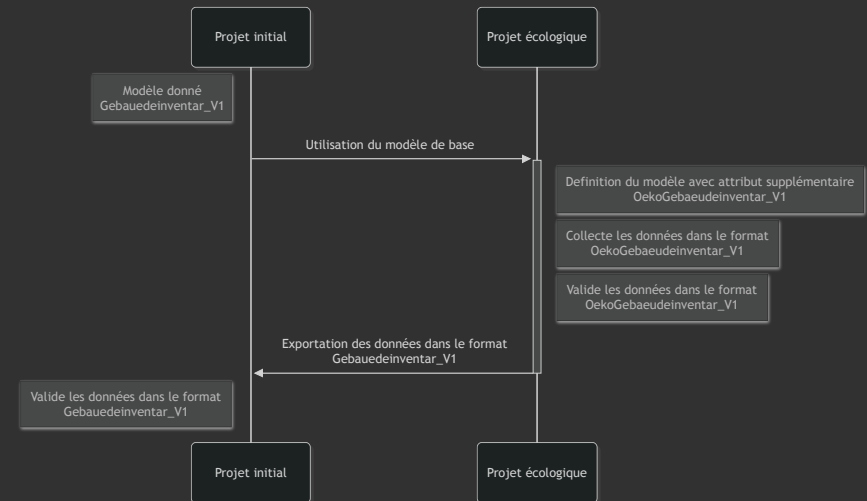
ET VOILÀ, NOUS L'AVONS !

La dérivation en INTERLIS (1)

... avec `EXTENDED`

```
1 CLASS Gebaeude =  
2   EGID : MANDATORY TEXT*16;  
3   Koordinaten : MANDATORY GeometryCHLV95_V1.  
4 END Gebaeude;
```

```
1 CLASS Gebaeude (EXTENDED) =  
2   EntsprichtStandard: BOOLEAN;  
3 END Gebaeude;
```



MAIS IL Y A ENCORE PLUS

Nous voulons maintenant saisir plusieurs types de bâtiments :

- Bâtiments résidentiels (avec `Anzahl_Wohnungen`)
- Bâtiments commerciaux (avec `Anzahl_Firmen`)
- Bâtiments commerciaux publics (avec `Anzahl_Firmen` et `Zweck`)

... et tout renvoyer finalement comme `Gebaeude`.

VÉRIFIEZ !

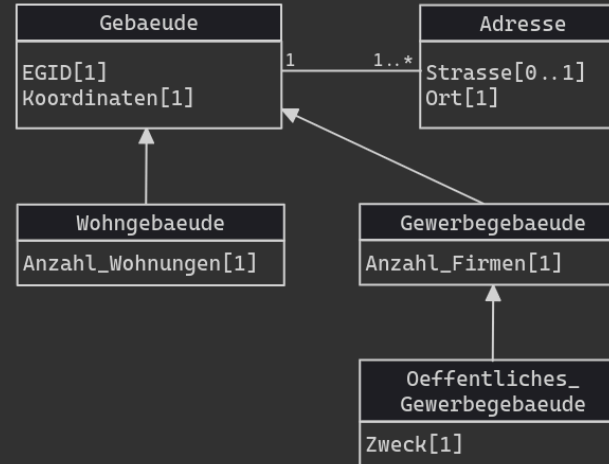
EXEMPLE DE MODÈLE MultiGebaeudeinventar_V1

La dérivation en INTERLIS (partie 2)

... avec EXTENDS

```
1 CLASS Gebaeude =
2   EGID : MANDATORY TEXT*16;
3   Koordinaten : MANDATORY GeometryCHLV95_V1;
4 END Gebaeude;
```

```
1 CLASS Gewerbegebaeude
2 EXTENDS Gebaeude =
3   Anzahl_Firmen : MANDATORY 1 .. 100;
4 END Gewerbegebaeude;
```



Les classes en INTERLIS

```
1 MODEL Gebaeudeinventar_V1 (de) =
2
3   TOPIC Gebaeude =
4
5     CLASS Gebaeude =
6       EGID : MANDATORY TEXT*16;
7       Koordinaten : MANDATORY GeometryCHLV95_V1;
8     END Gebaeude;
9
10    CLASS Adresse =
11      Strasse : TEXT*200;
12      Ort : MANDATORY TEXT*200;
13    END Adresse;
```

Les classes en INTERLIS

```
1 MODEL Gebaeudeinventar_V1 (de) =
2
3   TOPIC Gebaeude =
4
5     CLASS Gebaeude =
6       EGID : MANDATORY TEXT*16;
7       Koordinaten : MANDATORY GeometryCHLV95_V1;
8     END Gebaeude;
9
10    CLASS Adresse =
11      Strasse : TEXT*200;
12      Ort : MANDATORY TEXT*200;
13    END Adresse;
```

Les classes en INTERLIS

```
8      END Gebaeude;
9
10     CLASS Adresse =
11       Strasse : TEXT*200;
12       Ort : MANDATORY TEXT*200;
13     END Adresse;
14
15     ASSOCIATION AdresseGebaeude =
16       GebaeudeAdresse -- {1..*} Adresse;
17       Gebaeude -- {1} Gebaeude;
18     END AdresseGebaeude;
19
20   END Gebaeude;
```

Les classes en INTERLIS

```
18      END AdresseGebaeude;
19
20     END Gebaeude;
21
22   END Gebaeudeinventar_V1.
23
24   MODEL MultiGebaeudeinventar_V1 (de) =
25     IMPORTS Gebaeudeinventar_V1;
26
27     TOPIC Gebaeude EXTENDS Gebaeudeinventar_
28
29       !! First inheritance
30     CLASS Wohngebaeude
```

Les classes en INTERLIS

```
25     IMPORTS Gebaeudeinventar_V1;
26
27     TOPIC Gebaeude EXTENDS Gebaeudeinventar_
28
29       !! First inheritance
30     CLASS Wohngebaeude
31     EXTENDS Gebaeude =
32       Anzahl_Wohnungen : MANDATORY 1 .. 1000
33     END Wohngebaeude;
34
35       !! Second inheritance
36     CLASS Gewerbegebaeude
37     EXTENDS Gebaeude =
```

Les classes en INTERLIS

```
31     EXTENDS Gebaeude =
32       Anzahl_Wohnungen : MANDATORY 1 .. 1000
33     END Wohngebaeude;
34
35       !! Second inheritance
36     CLASS Gewerbegebaeude
37     EXTENDS Gebaeude =
38       Anzahl_Firmen : MANDATORY 1 .. 100;
39     END Gewerbegebaeude;
40
41       !! Inheritance of first inheritance
42     CLASS Oeffentliches_GewerbeGebaeude
43     EXTENDS Gewerbegebaeude =
```

Les classes en INTERLIS

```
36 CLASS Gewerbegebaeude
37 EXTENDS Gebaeude =
38   Anzahl_Firmen : MANDATORY 1 .. 100;
39 END Gewerbegebaeude;
40
41 !! Inheritance of first inheritance
42 CLASS Oeffentliches_GewerbeGebaude
43 EXTENDS Gewerbegebaeude =
44   Zweck : MANDATORY TEXT*255;
45 END Oeffentliches_GewerbeGebaude;
46 END Gebaeude;
47
48 END MultiGebaudeinventar_V1.
```

PRÊT POUR L'EXERCICE 2 🚀

INTERMÈDE 2.2 : REPRÉSENTATION PHYSIQUE DES HÉRITAGES

D'ABORD

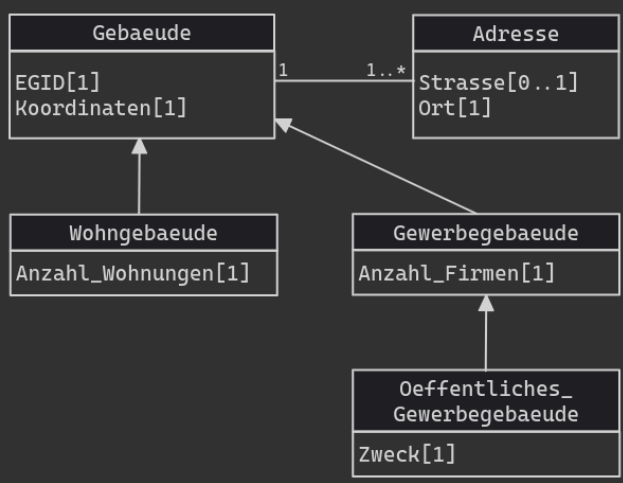
*Essayez avec Smart2 et si ça ne va pas,
essayez avec Smart1.*

Citation : Romedi Filli - 2022

STRATÉGIES DE MAPPING D'HÉRITAGE

- New Class
- Super Class
- Sub Class
- New+Sub Class

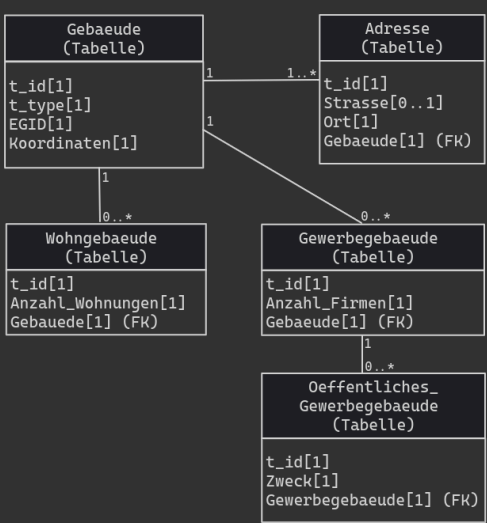
Exemple de modèle



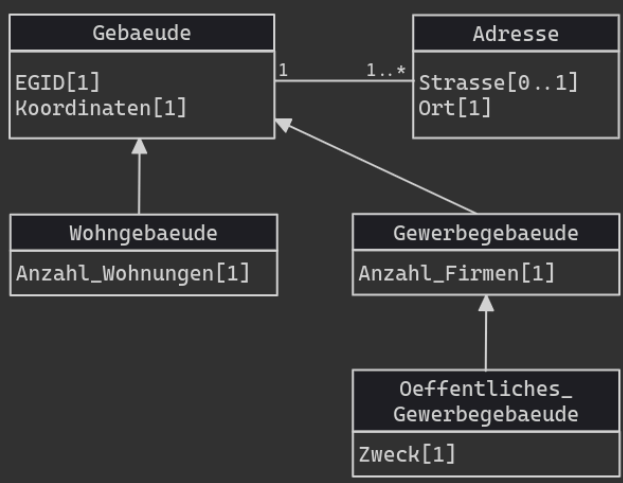
New Class

```
Gebaeude.t_type : (
    Wohngebaeude,
    Gewerbegebaeude,
    Oeffentliches_Gewerbegebaeude
)
```

- Les extensions sont mappées avec des relations.
- La condition NON NUL est respectée.
- Plusieurs importations sont nécessaires par objet.



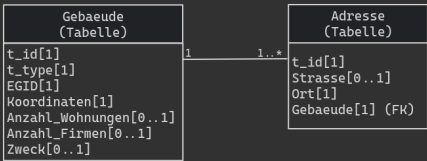
Exemple de modèle



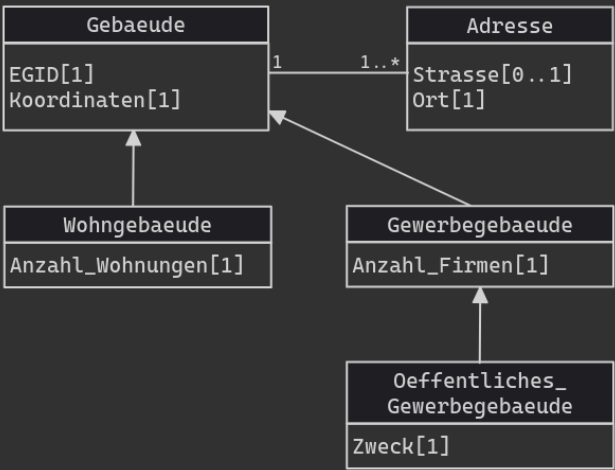
Super Class

```
Oeffentliches_Gewerbebaeude.t_type: (
    Gewerbebaeude,
    Oeffentliches_Gewerbebaeude
)
```

- Moins de tables et de relations (facile à utiliser)
- La condition NOT NULL n'est pas respectée



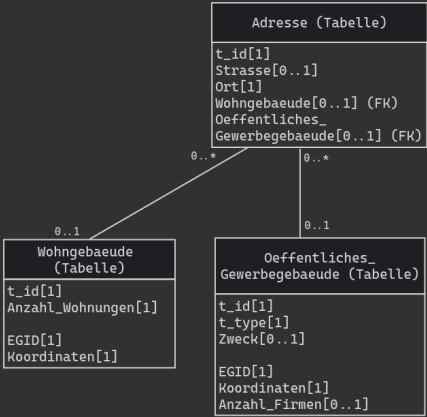
Exemple de modèle



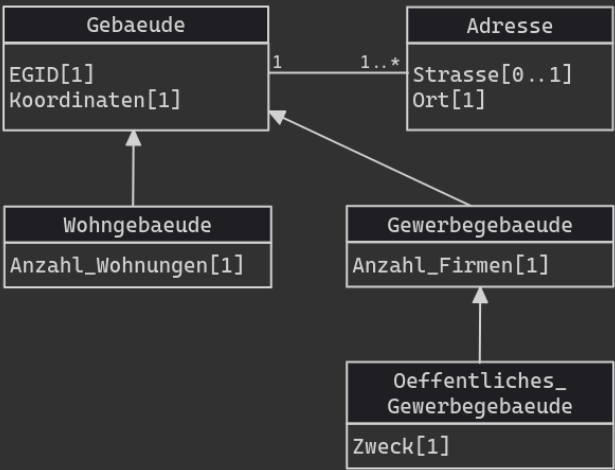
Sub Class

```
Oeffentliches_Gewerbebaeude.t_type: (
    Gewerbebaeude,
    Oeffentliches_Gewerbebaeude
)
```

- La condition NON NUL n'est pas respectée.

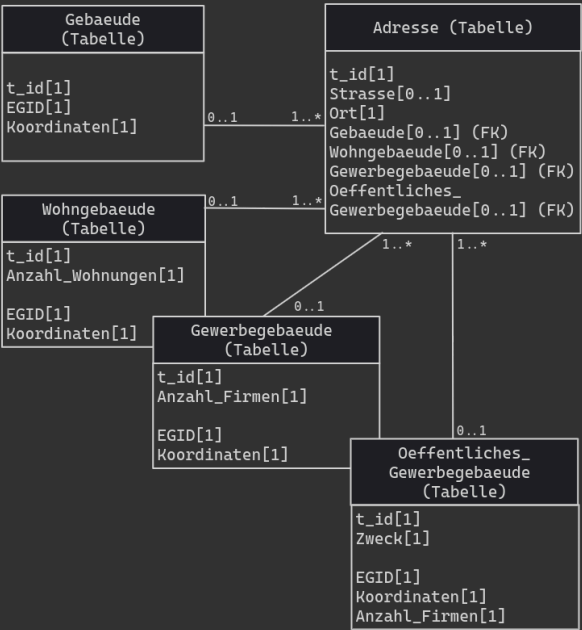


Exemple de modèle



New+Sub Class

- Pas de types
- À mon avis, la procédure attendue
- La condition NOT NULL est respectée



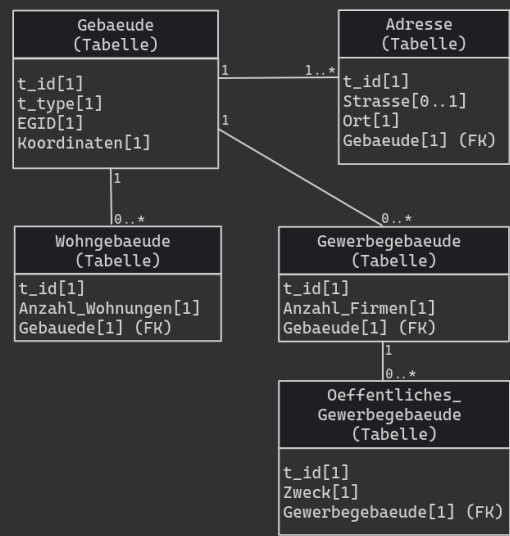
SMART MAPPING DANS ILI2DB

noSmartMapping

- Toutes les classes sont mappées avec la stratégie *New Class*.

noSmartMapping

Résulte en ce que nous voyons dans le graphique *New Class*.



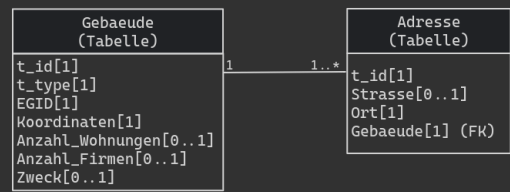
smart1Inheritance

- Classes abstraites sans relations -> stratégie *Super Class*
- Classes abstraites avec relations et sans super classe concrète -> stratégie *New Class*
- Classes concrètes sans super classe concrète -> stratégie *New Class*
- Toutes les autres classes -> stratégie *Super class*

smart1Inheritance

Signifie donc que Gebaeude utilise la stratégie *New Class* et toutes les autres la stratégie *Super class*.

Résulte en ce que nous voyons dans le graphique *Super class*.



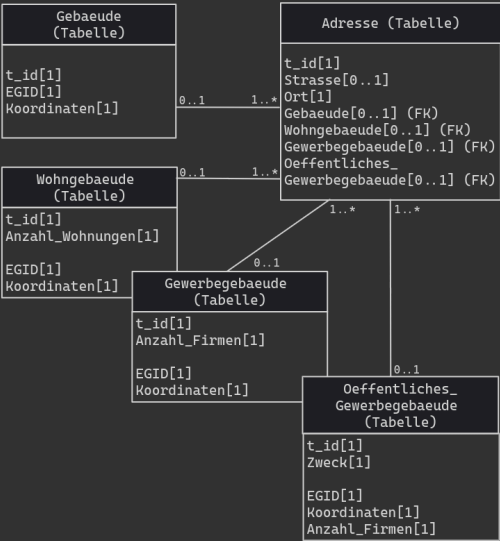
smart2Inheritance

- Classes abstraites -> stratégie *Super Class*
- Toutes les classes concrètes -> stratégie *New+Sub Class*

smart2Inheritance

Signifie donc que toutes utilisent la stratégie *New+Sub Class*.

Résulte en ce que nous voyons dans le graphique *New+Sub Class*.



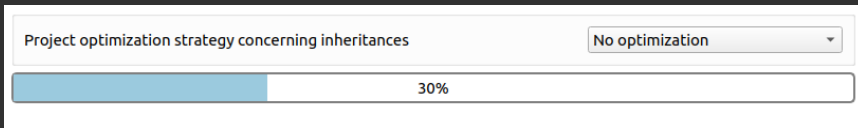
En résumé

- smart 1
- smart 2

IL EST TEMPS DE FAIRE L’EXERCICE 2.1 ? 🚀

INTERMÈDE 2.3 : OPTIMISATION DE PROJET AVEC SMART2

CRÉONS UN PROJET SANS OPTIMISATION



The screenshot shows a dialog box titled "Project optimization strategy concerning inheritances". It features a dropdown menu currently set to "No optimization" and a progress bar below it. The progress bar is partially filled with blue and is labeled "30%" on the right side.

... essayons avec `OekoGebaudeinventar_V1` (ou aussi `ZG_Nutzungsplanung_V1_1`)

POUVONS-NOUS SIMPLEMENT IGNORER LES CLASSES DE BASE ?

PROBLÉMATIQUE

Si un modèle ou un topic contient des classes étendues, celles-ci sont implémentées dans la base de données physique, y compris les classes de base...

... et par conséquent, des couches sont créées dans QGIS.

MAIS QUELLES COUCHES VOULONS-NOUS VOIR ?

- Les classes de base avec des extensions du même nom sont *non pertinentes*
- Les classes de base avec plusieurs extensions sont *non pertinentes*
- Sauf si l'extension est dans le même modèle, elles ne sont pas *non pertinentes*

STRATÉGIES

- Masquer les classes de base non pertinentes
- Grouper les classes de base non pertinentes
- Aucune : les couches sont renommées pour être uniques

VÉRIFIEZ !

PRÊT POUR LE DÎNER 

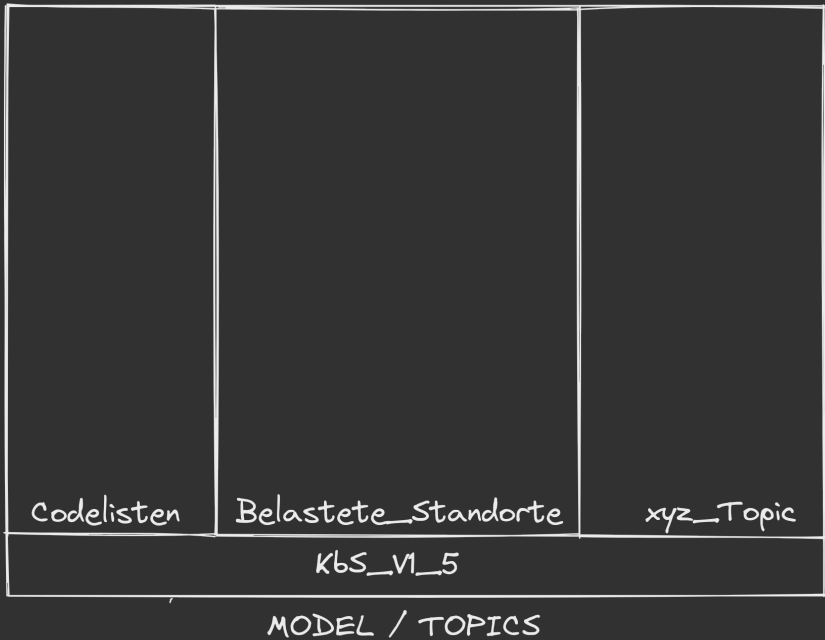
**INTERMÈDE 3 : CONTENEURS ET
ENSEMBLES DE DONNÉES**

ENSEMBLES DE DONNÉES (DATASETS)

- Ensembles de données d'un domaine **spatial ou thématique** spécifique
- **Indépendant** du modèle (contrairement au topic)
- Les données d'un ensemble de données peuvent être **gérées, validées et exportées** indépendamment des autres données.

CONTENEURS (BASKETS)

- Une instance plus petite
- Alors que les ensembles de données couvrent généralement l'ensemble du modèle, les conteneurs font généralement partie d'un topic
- Le plus souvent, ils sont l'intersection du topic et de l'ensemble de données



DATASETS

SH			
	Codelisten	Belastete_Standorte	xyz_Topic
	KbS_V1_5		
	MODEL / TOPICS		

DATASETS

TG			
SH			
	Codelisten	Belastete_Standorte	xyz_Topic
	KbS_V1_5		
	MODEL / TOPICS		

DATASETS

Catalogues			
TG			
SH			
Codelisten			Belastete Standorte
			xyz_Topic
			KbS_VI_5
			MODEL / TOPICS

DATASETS

Catalogues			
TG			
SH			
Codelisten			Belastete Standorte
			xyz_Topic
			KbS_VI_5
			MODEL / TOPICS

DATASETS

Catalogues	BASKET 5		
TG		BASKET 3	BASKET 4
SH		BASKET 1	BASKET 2
	Codelisten	Belastete_Standorte	xyz_Topic
	KbS_VI_5		

MODEL / TOPICS

PRÊT À DÉMARRER 🚀

INTERMÈDE 4 : QUE SONT LES OI D ?

OID

Les identifiants d’objet (OID) sont des chaînes de caractères **uniques à travers les systèmes** qui identifient un objet INTERLIS.

Pour **échanger** et **mettre à jour** des objets **sans conflit** à travers différents endroits.

TID

Dans le fichier de transfert, l’OID est visible sous TID.

```
<City_V1.Constructions.Buildings TID="chMBAKER00000100">  
  <Street>Rue des Fleures</Street><Number>1</Number>  
</City_V1.Constructions.Buildings>
```

POURQUOI TID ET PAS OID ?

- Un OID dans un XTF est un TID
- Un TID dans un XTF n’est *pas toujours* un OID

Car il n’est pas obligatoire de travailler avec des OID. Un fichier XTF a besoin de TID pour les références, etc.

Si vous n’utilisez pas d’OID, les TID sont simplement des nombres ou des caractères quelconques (pas de stabilité à travers les systèmes garantie !)

ET DANS LE SCHÉMA DE BASE DE DONNÉES PHYSIQUE ?

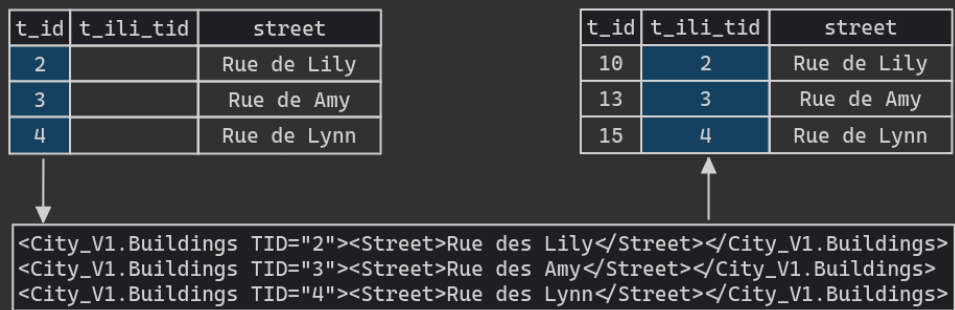
	t_id	t_ili_tid	Street	Number
1	3	chMBAKER00000100	Rue des Fleures	1
2	4	chMBAKER00000101	Rue des Fleures	2

t_id vs. t_ili_tid

- Le t_ili_tid correspond au TID dans le XTF et donc à l’OID (si des OID sont utilisés)
- Le t_id n’est qu’un numéro de séquence interne au système

Quand le t_id devient-il TID et le TID devient-il t_ili_tid ?

- Si le t_ili_tid reste vide, alors lors d’une exportation, le t_id est écrit dans le TID.
- Lors de l’importation, le TID est alors écrit dans le t_ili_tid.



QUAND DOIS-JE UTILISER DES OID ?

- Dès que plusieurs endroits collectent et gèrent des données
- Si le modèle le requiert :

```
[...]
TOPIC Constructions =
    BASKET OID AS INTERLIS.UUIDOID;
    OID AS INTERLIS.STANDARDROID;
[...]
```

TYPES D’OID

- UUIDOID
- I32OID
- STANDARDOID
- ANYOID
- OID défini par l’utilisateur

I32OID

- `OID 0 ... 2147483647`

Valeur par défaut Model Baker

```
t_id
```

UUIDOID

- `OID TEXT*36`
- Identificateur Unique Universellement

Valeur par défaut Model Baker

```
uuid( 'WithoutBraces' )
```

STANDARDOID

- `OID TEXT*16`
- **Préfixe (2 + 6 caractères)**

Identifiant du pays + une partie d’identifications *globale*.

- **Postfix (8 caractères)**

Séquence (numérique ou alphanumérique) du système comme partie d’identifications *locale*.

Valeur par défaut Model Baker

```
'%change%' || lpad( T_Id, 8, 0 )
```

ANYOID

Ne définis pas de format pour l’OID, mais seulement qu’un OID doit être défini dans tous les modèles Symbolisations étendus.

OID non définis

Les règles du type XML-ID (www.w3.org/TR/REC-xml) s’appliquent

- le premier caractère doit être une lettre ou un souligné
- suivi de lettres, chiffres, points, tirets, soulignés - rien d’autre.

Valeur par défaut dans Model Baker

```
'_' || uuid('WithoutBraces')
```

La même chose s’applique aux OID définis par l’utilisateur.

GESTIONNAIRE D’OID

Base de données > Model Baker > Gestionnaire d’OID

OID Manager

Define the default expression values used for the t_ili_tid field in the attribute forms according to the OID definition from the INTERLIS model.

Layer	OID Type	Default Value Expression	Show
Spaß.Gebaeude (OIDMadness_V1.Spaß)	INTERLIS.I32OID	123 T_Id	☐
Parkplatz (OIDMadness_V1.Business)	INTERLIS.STANDARDOID	00000' lpad(T_Id, 8, 0)	☒
Business.Gebaeude (OIDMadness_V1.Business)	INTERLIS.STANDARDOID	00000' lpad(T_Id, 8, 0)	☒
Quartier.Gebaeude (OIDMadness_V1.Quartier)	INTERLIS.UUIDOID	uuid("WithoutBraces")	☐
Fluss (OIDMadness_V1.Natur)	OIDMadness_V1.TypeIDShort	' uuid("WithoutBraces")	☐
See (OIDMadness_V1.Natur)	OIDMadness_V1.TypeID	" uuid("WithoutBraces")	☐
Wald (OIDMadness_V1.Natur)	INTERLIS.UUIDOID	uuid("WithoutBraces")	☐
BesitzerIn ...	INTERLIS.ANYOID	' uuid("WithoutBraces")	☐

☐ Reset the technical sequence used as t_id value

T_id (Sequence) Value

15

☐ Use logins from settings (superuser) for this operation

VÉRIFIEZ !

INTERMÈDE 5 : USABILITY HUB

À QUOI SERT USABILITY HUB ?

- Obtenir des informations supplémentaires sur les **modèles INTERLIS** et leurs **projets (QGIS)** automatiquement via le web.
- Configurations supplémentaires faites **une seule fois** et utilisées plusieurs fois.

COMMENT FONCTIONNE USABILITY HUB ?

1. L'utilisateur sélectionne le modèle

- Recherche inter-référentiels de **fichiers de méta-configuration**.
- Trouver les **fichiers de méta-configuration** dans les `ilidata.xml` sur UsabilityHub (<https://models.opengis.ch>) ou les référentiels propres.

2. Le fichier de méta-configuration est analysé

Les **configurations** pour ili2db, Model Baker et d’autres outils y sont définies.

Également des **liens** (lds) vers :

- Catalogues
- Fichier de méta-attributs supplémentaire
- Fichiers de surcouche

3. Les configurations sont définies

- ili2db : Smart-Mapping, gestion des conteneurs, etc.

4. Les liens sont résolus

- Recherche inter-référentiels des **catalogues, fichier de méta-attributs supplémentaire et fichiers de surcouche**

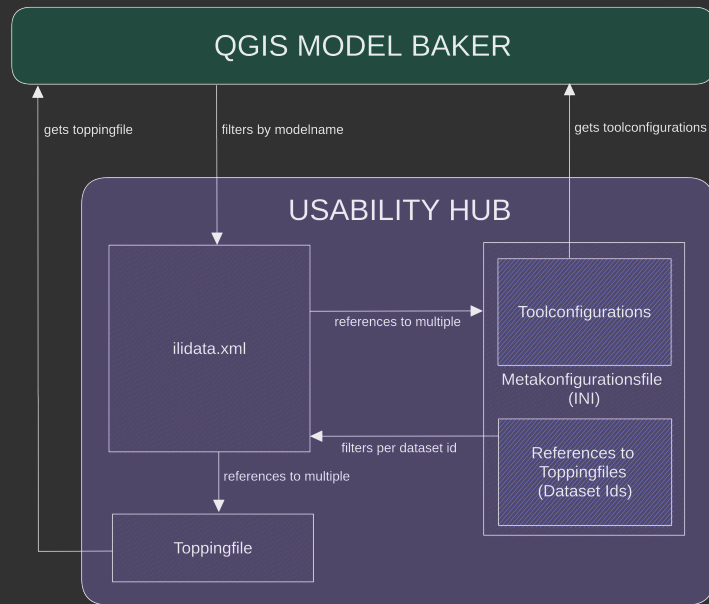
Les **fichiers de surcouche** contiennent des informations sur :

- Symbolologies et configurations de formulaires
- Disposition des légendes et ordre des couches
- Thèmes de carte, mises en page d’impression
- etc.

5. Utilisation des informations

- Exécuter ili2db en utilisant les configurations et le fichier de méta-attributs supplémentaire
- Importer les catalogues trouvés
- Générer le projet en utilisant les fichiers de surcouche

DÉROULEMENT



DOCUMENTATION

Voir <https://usabilityhub.opengis.ch/>

Ou la [documentation de Model Baker](#)

PRÊT À DÉMARRER 🚀

INTERMÈDE 6 : MODÈLES DE TRADUCTION

QUE SONT LES MODÈLES DE TRADUCTION ?

Les modèles de traduction sont des *modèles traduits* qui ne diffèrent *pas* du modèle original en termes de structure.

EXEMPLE GEBAEUDEINVENTAR_V1

<pre>MODEL Gebaeudeinventar_V1 (de) AT "http://modelbaker.ch" VERSION "2024-08-08" = IMPORTS GeometryCHLV95_V1; TOPIC Bauwerk = OID AS INTERLIS.UUIDOID; DOMAIN Kanton = (AG, AI, AR, BE, BL, BS, FR, GE, GL, GR, JU, LU, NE, NW, OW, SG, SH); CLASS Gebaeude = EGID : MANDATORY TEXT*16; Kantonskuerzel : MANDATORY Kanton; Koordinaten : MANDATORY GeometryCHLV95_V1.Coord2; Bauperiode : MANDATORY TEXT; Flaeche : 0.00 .. 99999.99; Geschosse : 0 .. 999; Zivilschutzraum : MANDATORY BOOLEAN; Heizung : TEXT; Datum_Heizung : INTERLIS.XMLDate; UNIQUE EGID; SET CONSTRAINT WHERE DEFINED (Heizung) : DEFINED (Datum_Heizung); END Gebaeude; END Bauwerk;</pre>	<pre>MODEL InventaireBatiments_V1 (fr) AT "http://modelbaker.ch" VERSION "2024-09-24" TRANSLATION OF Gebaeudeinventar_V1 ["2024-08-08"] = IMPORTS GeometryCHLV95_V1; TOPIC Construction = OID AS INTERLIS.UUIDOID; DOMAIN Canton = (AG, AI, AR, BE, BL, BS, FR, GE, GL, GR, JU, LU, NE, NW, OW, SG); CLASS Batiment = EGID : MANDATORY TEXT*16; CantonIdentifiant : MANDATORY Canton; Coordonnees : MANDATORY GeometryCHLV95_V1.Coord2; PeriodeDeConstruction : MANDATORY TEXT; Surface : 0.00 .. 99999.99; Etages : 0 .. 999; AbriDeProtection : MANDATORY BOOLEAN; Chauffage : TEXT; Date_Chauffage : INTERLIS.XMLDate; UNIQUE EGID; SET CONSTRAINT WHERE DEFINED (Chauffage) : DEFINED (Date_Chauffage); END Batiment; END Construction;</pre>
---	---

PRÊT À DÉMARRER 🚀

EXEMPLE PLANS D'AFFECTATION

Fr : https://models.geo.admin.ch/ARE/PlansDAffectation_V1_2.ili

De : https://models.geo.admin.ch/ARE/Nutzungsplanung_V1_2.ili

PRÊT À DÉMARRER 🚀

EXERCICES

... la pratique rend parfait

EXERCICE 1 : CRÉER UN PROJET UTILISABLE

- Créer un projet pour `KbS_V1_5` incluant un fichier de méta-attributs supplémentaire
- Ajouter une carte de fond
- Enregistrer le projet

EXERCICE 2 : SMART INHERITANCE MAPPING

- Importer le modèle `MultiGebaeudeinventar_V1` avec **Smart1Inheritance**
- Renommer les couches en `<nom de la couche>_Smart1`
- Importer le modèle `MultiGebaeudeinventar_V1` avec **Smart2Inheritance**

Ne pas utiliser d'optimisation :

Project optimization strategy concerning inheritances

No optimization

30%

EXERCICE 2.1 : OPTIMISATION DU PROJET

- Importer le modèle `0ekoGebaeudeinventar_V1` avec **Smart2Inheritance**
- Importer le modèle `ZG_Nutzungsplanung_V1_1` avec **Smart2Inheritance**

Ne pas utiliser d'optimisation :

Project optimization strategy concerning inheritances

No optimization

30%

EXERCICE 3 : SAISIE DANS LES ENSEMBLES DE DONNÉES

- Créer le projet avec des ensembles de données pour TG et SH (si ce n'est pas déjà fait)
- Ajouter une zone à Stein am Rhein TG.
- Saisir une zone dans SH.
- Créer un nouvel ensemble de données pour ZH.
- Saisir une zone dans ZH.

EXERCICE 4 : EXPORTATION

- Exporter toutes les données dans un fichier de transfert (y compris les listes de codes)
- Exporter uniquement les données utilisateur de SH dans un fichier de transfert

TÂCHE SUPPLÉMENTAIRE

- Créer un projet pour `0ekoGebaeudeinventar_V1` avec Smart2 et l'optimisation de masquage
- Saisir un objet
- Exporter pour `0ekoGebaeudeinventar_V1` et pour `Gebaeudeinventar_V1`

EXERCICE 5 : USABILITYHUB

- Configurer un projet quelconque
- Exporter les surcouches
- Importer le modèle avec les surcouches du référentiel local



ÉQUIPE DE PROJET

par ordre d'apparition

- Romedi Filli, Canton SH (direction)
- Manuel Kaufmann, Agroscope
- David Signer, [OPENGIS.ch](https://www.opengis.ch)
- Pascal Megert, Canton AI
- Marleen Schulze, Canton SZ
- Adrian Weber, Canton SO

Un groupe de différents intéressés...

MODEL BAKER GROUP

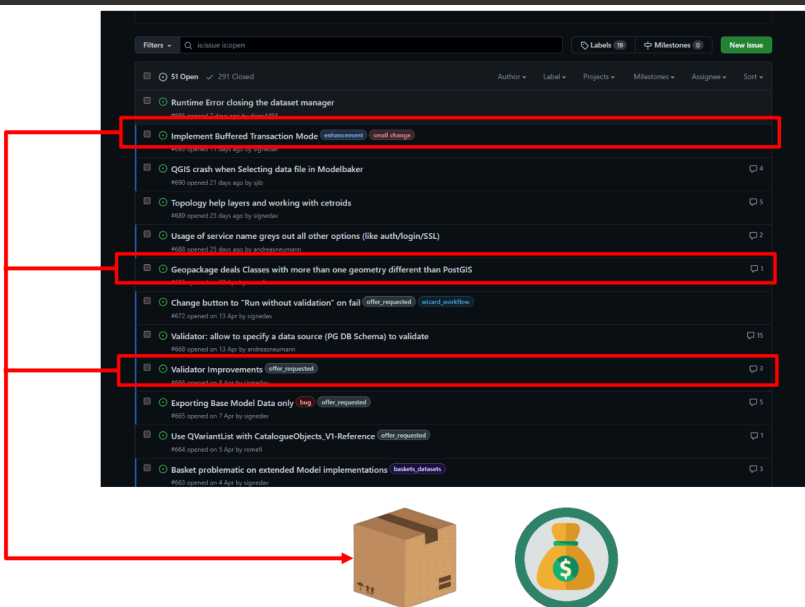
CONTRIBUTION ANNUELLE AVEC DES NIVEAUX

(à partir du parrain avec heures de support incluses) :

- Visionnaire
- Parrain
- Soutien
- Donateur

FINANCEMENT

PAR PROJET



```
23-01-19" =  
GeometryCHLV95_V1, CatalogueObjects_V1, LocalisationCH_V1;
```

```
alog =
```

```
heizung_Catalogue  
CatalogueObjects_V1.Catalogues.Item =  
: MANDATORY TEXT;  
reibung : MANDATORY LocalisationCH_V1.MultilingualText;  
zung_Catalogue;
```

```
RE : Heizung_Catalogue  
CatalogueObjects_V1.Catalogues.Item =  
enc : MANDATORY REFERENCE TO (EXTERNAL) Heizung_Catalogue;  
zung_Catalogue;  
og;
```

```
aeude  
ON : baeude : Heizung_Catalogue;
```

```
ton  
AG, AI, AR, BE, BL, BS, FR, GE, GL, GR, JU, LU, NE, NW, OW, SG, SH, SO, SZ, TG, TI, UR, VS, ZG, ZH
```

```
ebaeude =  
D : MANDATORY TEXT*8;  
tonskunzel : MANDATORY Kanton;  
ndstuecknummer : MANDATORY 0 .. 99999;  
e : TEXT;  
rdinaten : MANDATORY GeometryCHLV95_V1.Coord2;  
tus : MANDATORY TEXT;
```

DES QUESTIONS

?

C'EST TOUT...

Web : www.opengis.ch

Courriel : david@opengis.ch

Github : github.com/opengisch / github.com/signedav

Mastodon : [@opengisch@fosstodon.org](https://fosstodon.org/@opengisch) /
[@signedav@fosstodon.org](https://fosstodon.org/@signedav)