








# INTERLIS KONKRET: QGIS MODEL BAKER



Tobias Heini, Pia Bereuter

# DIE APPLIKATIONEN RUND UM INTERLIS

Name	Funktion	Weitere Infos
 Ili2 Validator	Datenvalidierung: Prüft INTERLIS-Daten auf deren Richtigkeit	<a href="#">Link</a>
 ili2c	Kompilieren/ Prüfung von Modellen	<a href="#">Link</a>
 Ili2db	Datenbanktool: ...erstellen von Datenbank-Schemen auf Basis eines Modelles ...importieren und exportieren von Daten	<a href="#">Link</a>
 Ili2fme	Erweiterung für FME: ermöglicht das Lesen und Schreiben von INTERLIS Daten mithilfe von FME	<a href="#">Link</a>
 ILICOP	Onlineapplikation für die Datenvalidierung auf Basis des Ili2 Validator	<a href="#">Link</a>
 Model Browser	Onlineapplikation für das Finden von Modellen	<a href="#">Link</a>
 ModelBaker	QGIS-Erweiterung für... -das Erstellen eines QGIS-Projektes zur Erfassung von Daten -den Import und Export von INTERLIS-Daten	<a href="#">Link</a>

# DIE BASIS DES QGIS MODEL BAKERS: ILI2DB

**ili2db** is ein Programm für das Datenbank-Management in Kombination INTERLIS

ili2db funktioniert mit...

- Postgres: Ili2pg
- Filegeodatabase: Ili2fgd
- Geopackage: Ili2gpkg
- Oracle: Ili2ora
- Microsoft SQL-Server: Ili2mssql
- MySQL: Ili2mysql
- H2: Ili2h2gis

# FUNKTIONEN VON ILI2DB



Modell.ili



Schemaimport



Datenbankschema



Daten.xtf



Datenimport



Daten in der Datenbank



Daten in der Datenbank



Datenexport



Daten.xtf

## Demo

- Anleitung: <https://github.com/claeis/ili2db/blob/master/docs/ili2db.rst>

**WAS IST QGIS MODEL BAKER**

# EIN QGIS PROJEKTGENERATOR

Schnelles **Erstellen eines QGIS-Projekts** aus einem physischen Datenmodell.

Analysiert die vorhandene Struktur und konfiguriert ein QGIS-Projekt mit allen verfügbaren Informationen.

# EIN FÜR *INTERLIS* OPTIMIERTER QGIS PROJEKTGENERATOR

In INTERLIS definierte Modelle bieten **zusätzliche Metainformationen** wie Domänen, Einheiten von Attributen oder objektorientierte Definitionen von Tabellen.

Diese können genutzt werden, um die Projektkonfiguration weiter zu optimieren.

# EINE ILI2DB KONTROLLSTATION

```
java -jar /home/dave/dev/opengisch/QgisModelBaker/QgisModelBaker.jar --dbusr postgres --dbpwd ***** --dbdatabase bakery --dbschema bakery --createFk --createFkIdx --coalesceMultiSurface --coalesceMultiSurface --expandMultilingual --createTypeConstraint --createEnumTables --models Wildruhezonen_LV95_V2_1
```

# UND ES IST EINE LIBRARY

Kann als Framework genutzt werden.

... und ist auch als Python Package verfügbar:

```
pip install modelbaker
```

# ZIELE

- Vereinfachter Zugang/Umgang mit INTERLIS
- Bessere Datenqualität
- Automatisierung eines aufwendigen Prozesses

# ENTWICKLUNG

- OpenGIS: <https://opengisch.github.io/QgisModelBaker/de/>
- Gesteuert durch ein Projektteam mit Vertreter:innen aus SH, Agroscope, SZ, SO, AI

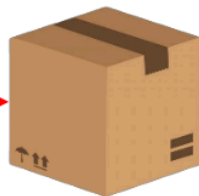
**FINANZIERUNG**

# PROJEKTBEZOGEN

Filters  Labels 19 Milestones 0 [New issue](#)

51 Open 291 Closed

Author	Label	Projects	Milestones	Assignee	Sort
<b>Runtime Error closing the dataset manager</b> #695 opened 7 days ago by dom1494					
<b>Implement Buffered Transaction Mode</b> <span>enhancement</span> <span>small change</span> #693 opened 11 days ago by signedav					
<b>QGIS crash when Selecting data file in Modelbaker</b> <span>4</span> #690 opened 21 days ago by sjib					
<b>Topology help layers and working with cetroids</b> <span>5</span> #689 opened 25 days ago by signedav					
<b>Usage of service name greys out all other options (like auth/login/SSL)</b> <span>2</span> #688 opened 25 days ago by andreasneumann					
<b>Geopackage deals Classes with more than one geometry different than PostGIS</b> <span>1</span> #685 opened 23 days ago by signedav					
<b>Change button to "Run without validation" on fail</b> <span>offer_requested</span> <span>wizard_workflow</span> #672 opened on 13 Apr by signedav					
<b>Validator: allow to specify a data source (PG DB Schema) to validate</b> <span>15</span> #668 opened on 13 Apr by andreasneumann					
<b>Validator Improvements</b> <span>offer_requested</span> <span>3</span> #666 opened on 8 Apr by signedav					
<b>Exporting Base Model Data only</b> <span>bug</span> <span>offer_requested</span> <span>5</span> #665 opened on 7 Apr by signedav					
<b>Use QVariantList with CatalogueObjects_V1-Reference</b> <span>offer_requested</span> <span>1</span> #664 opened on 5 Apr by romefi					
<b>Basket problematic on extended Model implementations</b> <span>baskets_datasets</span> <span>3</span> #663 opened on 4 Apr by signedav					





# JÄHRLICHER BEITRAG MIT ABSTUFUNGEN

(ab Förderer mit Supportstunden inklusive):

- Visionär
- Förderer
- Unterstützer
- Gönner

# LINKS

- Repository QGIS Plugin: <https://github.com/opengisch/qgisModelBaker>
  - Code
  - Issues
- Code Repository Backend Library: <https://github.com/opengisch/QgisModelBakerLibrary>
- Dokumentation: <https://opengisch.github.io/QgisModelBaker/>
  - Beschrieb, Anleitung usw.
- INTERLIS discourse group: <https://interlis.discourse.group>
  - Diskussionen
  - Allgemeine Infos
  - Software Informationen

**ÜBUNGEN**

# ÜBUNG: VORAUSSETZUNGEN

- QGIS Version 3.44.0 oder höher(<https://www.qgis.org/de/site/forumsers/download.html>)
- QGIS-Plugin “QGIS Model Baker”
- Internetzugang
- Optional: DB Browser([sqlitebrowser.org](https://sqlitedbbrowser.org))
- Java (alternativ Open JDK)

# JAVA

Habe ich Java installiert?

```
java --version
```

... kommt etwas, dann gut ist ☕

## Download für Windows / Mac

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

## Oder in Linux (openJDK)

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

or

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

# LERNZIELE

- Du kannst ein QGIS-Projekt mit dem “Model Baker” erstellen.
- Du kannst Daten über den “Model Baker” erfassen, in eine Geopackage schreiben und ins Format xtf exportieren.
- Du kannst Daten im Format xtf über den “Model Baker” in eine Geopackage importieren.
- Du kannst mit dem Model Baker Daten Validator Daten prüfen, die Meldungen interpretieren und die Fehler beheben.
- Du kennst weitere Applikationen um INTERLIS

# TIPPS UND TRICKS

- Ein guter File-Explorer:
  - FreeCommander <https://freecommander.com/de/ubersicht/>
- Shortcuts wie F6 für Tabelle öffnen
  - Objekt abfragen
  - Bearbeitungsmodus umschalten
  - Karte verschieben
- Nützliche QGIS-Plugins:
  - HCMGIS/QuickMapServices
  - Swiss Locator

# INFOS

## DATEI-ENDUNGEN:

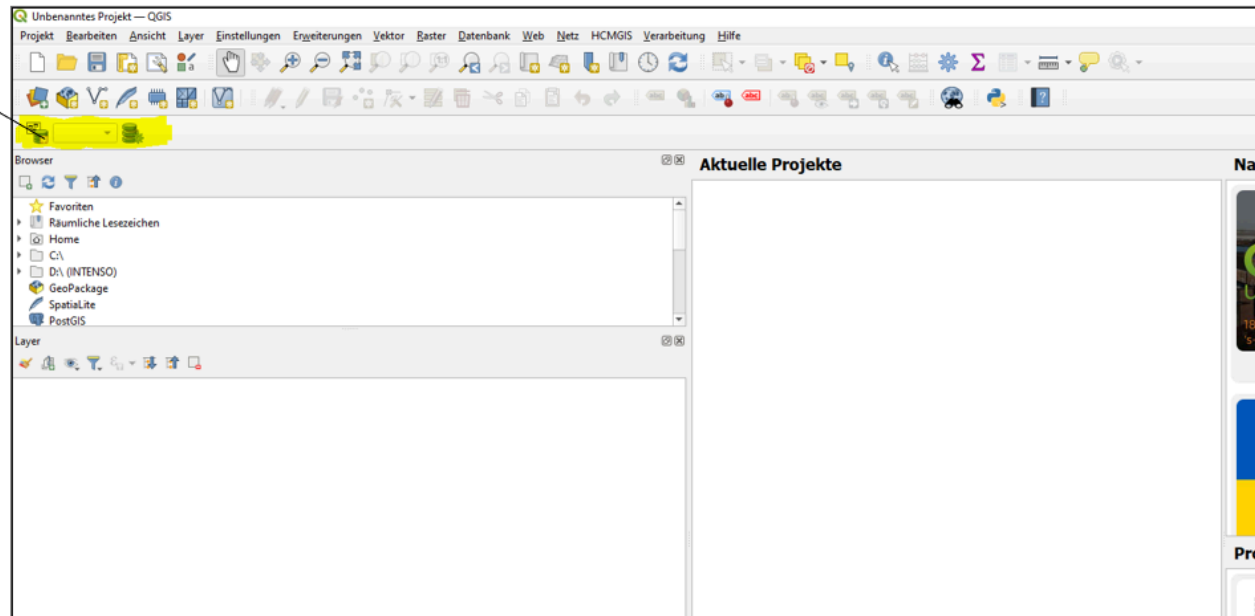
- **ili** Bezeichnung für *Modell Datei* (keine Daten)
- **itf** Bezeichnung der Transferdatei für Daten nach INTERLIS 1; steht für *INTERLIS-Transfer-File*
- **xtf** Bezeichnung der Transferdatei für Daten nach INTERLIS 2; steht für *XML-Transfer-File*

## DATEIEN FÜR DIE ÜBUNG

Anlagensanierung.ili	Datenmodell für die Übung zum Model Baker
SZ_Waermenutzung_V2.ili	Datenmodell von Standorten, an welchen Warmenutzungsanlagen erstellt wurden.
SZ_Waermenutzung_V2.xtf	Standorte von Warmenutzungsanlagen(folgt dem Modell "SZ_Waermenutzung_V2")
SZ_Waermenutzung_V2_fehlerhaft.xtf	Fehlerhafte Warmenutzungsflächen(folgt dem Modell "SZ_Waermenutzung_V2")

# START MODEL BAKER

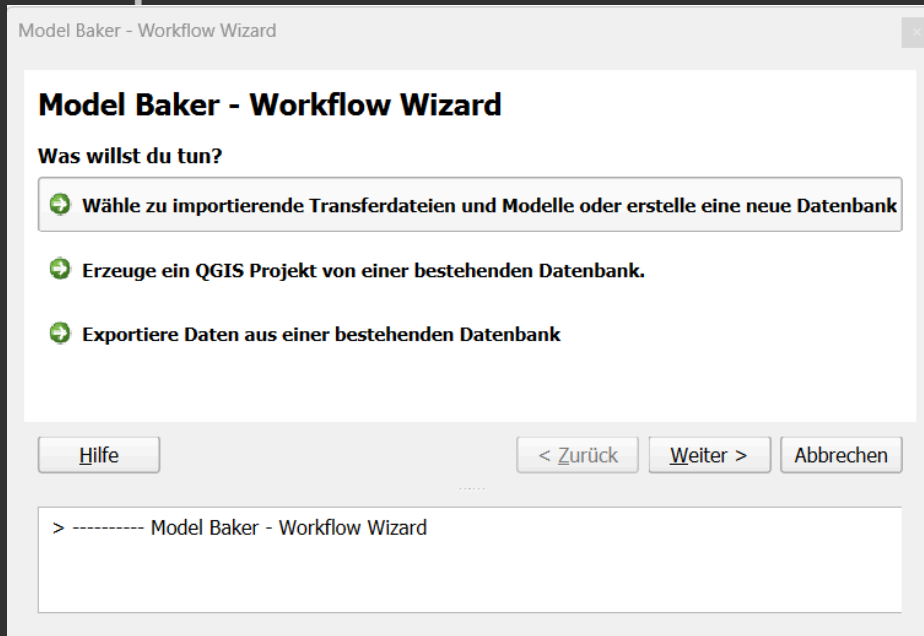
Nach der Installation des Plugins "Model Baker" erscheint eine Werkzeugleiste für den Modelbaker. Das linke Icon startet den Workflow Wizard der als Startpunkt für die wichtigsten Prozesse dient:



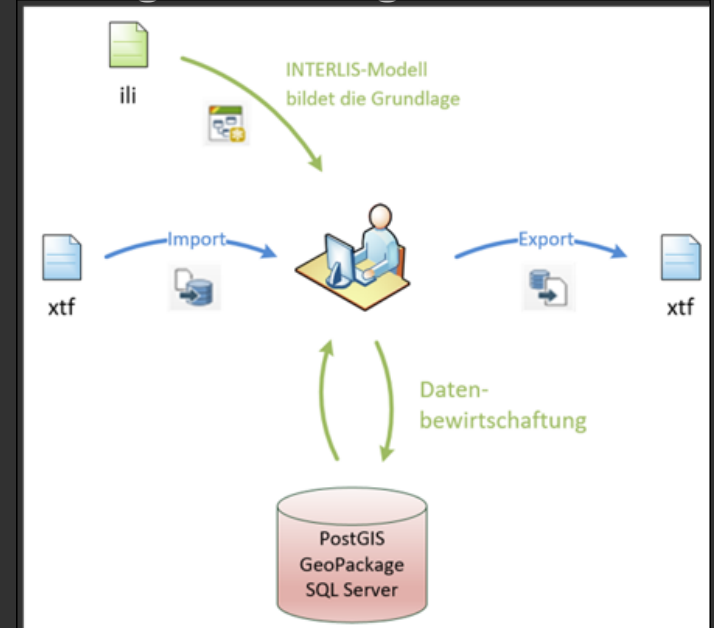


# ...FOLGENDE PROZESSE WERDEN DURCH DEN WORKFLOW WIZARD ERMÖGLICHT:

## Die Optionen des Wizards



## ermöglichen folgende Prozesse



Anleitung:

[https://opengisch.github.io/QgisModelBaker/user\\_guide/get\\_started/](https://opengisch.github.io/QgisModelBaker/user_guide/get_started/)

[https://opengisch.github.io/QgisModelBaker/user\\_guide/import\\_workflow/](https://opengisch.github.io/QgisModelBaker/user_guide/import_workflow/)

# FUNKTION: SCHEMA UND DATEN IMPORTIEREN

Die erste Option ist die zentrale Funktion des Model Bakers und läuft wie folgt ab:

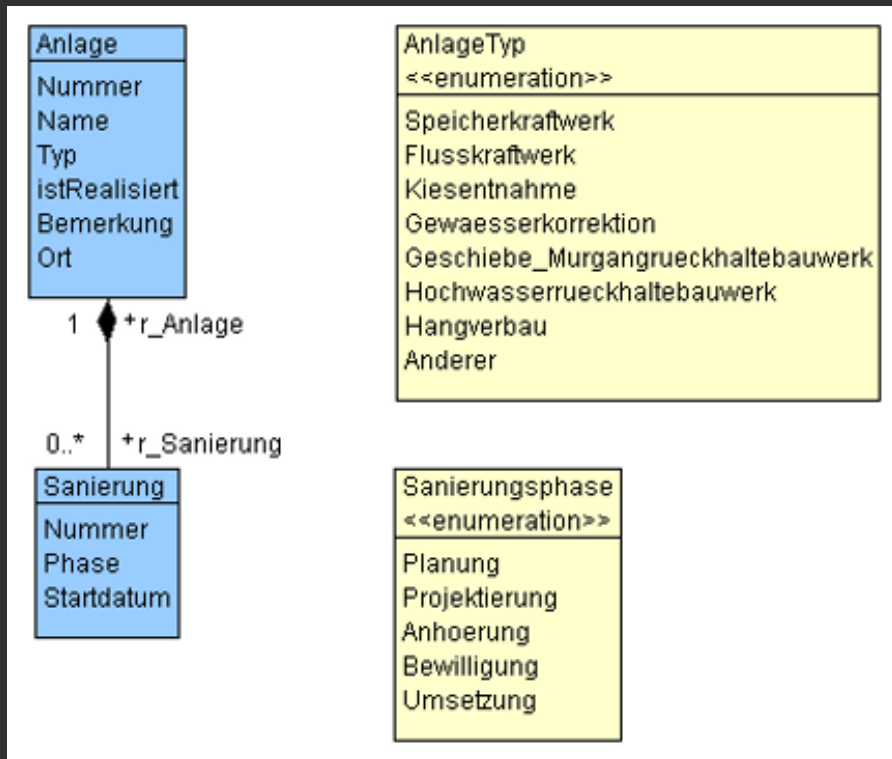
1. Auswählen des Modells aus welchem ein Datenbankschema generiert werden soll(optional), und/oder Transferfile(s) angeben, die in eine neue oder bestehende Datenbank integriert werden.
2. Auswählen einer Datenbank. Falls bei Schritt 1 ein Modell angegeben wurde, wird aus diesem eine Datenbank generiert.
3. Datenbankschema generieren
4. Datenimport konfigurieren(falls ein xtf angegeben wurde)
5. Datenimport Sessions durchführen(falls ein xtf ausgewählt wurde wurde)
6. Projekt erzeugen

# SPEZIALFUNKTIONEN

- **OID Manager**
  - Festlegen von der OID Typen und wie solche gesetzt werden
- **Daten Validator**
  - Daten validieren und Fehler anzeigen
- **Toppings**
  - Zusätzliche Styling Informationen für ein Projekt (Symbolisierung, Formulare etc.)
- **Spracheinstellungen**
  - Falls mehrere Sprachen für ein Modell verfügbar sind kann ausgewählt werden

 mehr dazu in den Exkursen

# DAS ÜBUNGSMODELL: ANLAGESANIERUNG



## KLASSEN

- **Anlage:**  
Tabelle mit einer Punktgeometrie welche die Lage von Anlagen ausweist
- **Sanierung:**  
Sanierungen die bei Anlagen ausgeführt wurden

## Domains

- **Anlage:**  
Typ welcher einer Anlage zugewiesen werden kann
- **Sanierung:**  
Sanierungsphase: Stadium, Phase der Sanierung; Auswahlliste für Sanierung

**START ÜBUNGEN**

# EXKURS 1: BEHÄLTER UND DATASETS

# DATASETS

- Datensätze eines bestimmten **räumlichen oder thematischen** Bereichs
- **Unabhängig** des Modells (im Gegensatz zum Topic)
- Die Daten eines Datasets können unabhängig von den anderen Daten **verwaltet, validiert und exportiert** werden.

## BASKETS (BEHÄLTER)

- Eine kleinere Instanz
- Währenddem die Datasets meist das ganze Modell umfassen, sind die Behälter meist Teil eines Topics
- Meistens sind sie die Schnittmenge von Topic und Dataset

Codelisten	Belastete Standorte	xyz_Topic
kbs_v1_5		

MODEL / TOPICS



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



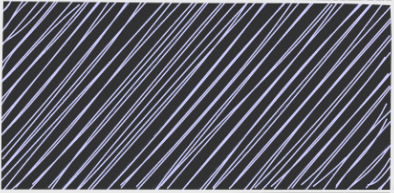
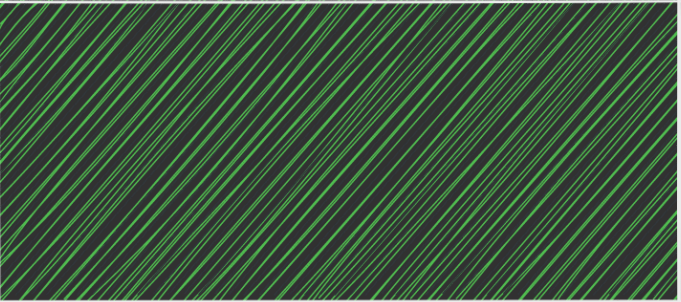
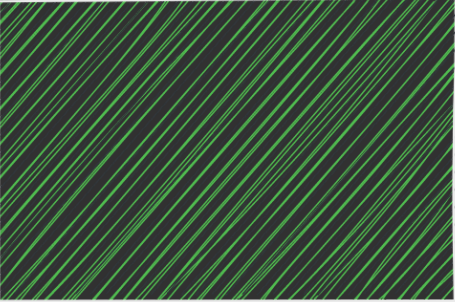

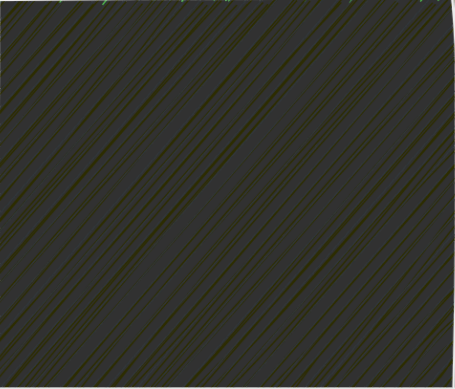
# DATASETS

Catalogues			
TG			
SH			
	Codelisten	Belastete_Standorte	xyz_Topic
	kbs_v1_5		

MODEL / TOPICS



DATASETS

Catalogues			
TG			
SH			
	Codelisten	Belastete_Standorte	xyz_Topic
	kbs_v1_5		

MODEL / TOPICS



DATASETS

Catalogues	BASKET 5		
TG		BASKET 3	BASKET 4
SH		BASKET 1	BASKET 2
	Codelisten	Belastete_Standorte	xyz_Topic
	kbs_v1_5		

MODEL / TOPICS



## **EXKURS 2: WAS SIND OIDS?**

## OIDs

Objekt-IDs (OID) sind **systemübergreifend eindeutige** Zeichenketten, die ein INTERLIS Objekt identifizieren.

Um Objekte **konfliktlos** über verschiedene Stellen **austauschen** und **updaten**.

## TIDS

In der Transferdatei ist die OID unter TID einsehbar.

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

## WESHALB TID UND NICHT OID?

- Eine OID ist im XTF eine TID
- Eine TID im XTF ist aber *nicht immer* eine OID

Denn man muss nicht unbedingt mit OIDs arbeiten. Eine XTF Datei braucht aber TIDs für Referenzen etc.

Wenn man keine OIDs benutzt, sind die TIDs einfach irgendwelche Nummern oder Zeichen (keine systemübergreifende Stabilität garantiert!)

## UND IM PHYSISCHEN DATENBANKSCHEMA?

	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

- Die `t_ili_tid` entspricht der **TID** im XTF und somit der **OID** (sofern OIDs verwendet werden)
- Die `t_id` ist nur eine **systeminterne** Sequenz-Nummer

## Wann wird die `t_id` zur TID und die TID zur `t_ili_tid`?

- die `t_ili_tid` leer bleibt, dann wird bei einem **Export** die `t_id` in die TID geschrieben.
- beim **Import** wird dann die TID in die `t_ili_tid` geschrieben.

t_id	t_ili_tid	street
2		Rue de Lily
3		Rue de Amy
4		Rue de Lynn


t_id	t_ili_tid	street
10	2	Rue de Lily
13	3	Rue de Amy
15	4	Rue de Lynn

<City\_V1.Buildings TID="2"><Street>Rue des Lily</Street></City\_V1.Buildings>  
<City\_V1.Buildings TID="3"><Street>Rue des Amy</Street></City\_V1.Buildings>  
<City\_V1.Buildings TID="4"><Street>Rue des Lynn</Street></City\_V1.Buildings>

## WANN MUSS/SOLL ICH OIDS VERWENDEN?

- Sobald mehr als eine Stelle Daten sammelt und verwaltet
- Wenn es das Modell voraussetzt:

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

 **Tipp:** OIDS immer verwenden: kein Mehraufwand, kann bei Datenverlust helfen und erleichtert die Nachvollziehbarkeit von Änderungen.

## ARTEN VON OIDS

- UUIDOID
- I32OID
- STANDARDOID
- ANYOID
- Benutzerdefinierte OID

## UUIDOID

- `OID TEXT*36`
- Universally Unique Identifier

## Standardwert Model Baker

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

## I32OID

- `OID 0 ... 2147483647`

## Standardwert Model Baker

```
t_id
```

## STANDARDOID

- `OID TEXT*16`

- **Präfix (2 + 6 Zeichen)**

Länderkennung + ein *globaler* Identifikationsteil.

- **Postfix (8 Zeichen)**

Sequenz (numerisch oder alphanumerisch) des Systems als *lokaler* Identifikationsteil.

## Standardwert Model Baker

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

## ANYOID

Definiert kein Format für die OID, sondern nur, dass eine OID in allen erweiterten Modellen definiert werden muss.

## Nicht definierte OIDs

Es gelten Regeln des **XML-ID-Typs** ([www.w3.org/TR/REC-xml](http://www.w3.org/TR/REC-xml))

- **erstes** Zeichen muss ein **Buchstabe oder Unterstrich** sein
- gefolgt von **Buchstaben, Zahlen, Punkten, Minuszeichen, Unterstrichen** - sonst nix.

## Standardwert im Model Baker

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

*Gleiches gilt für benutzerdefinierte OIDs.*

# OID MANAGER

Datenbank > Model Baker > OID Manager

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	<code>123 T_Id</code>	<input type="checkbox"/>
Parkplatz (OIDMadness_V1.Business)	INTERLIS.STANDARDROID	<code>00000'    lpad( T_Id, 8, 0 )</code>	<input checked="" type="checkbox"/>
Business.Gebaeude (OIDMadness_V1.Business)	INTERLIS.STANDARDROID	<code>00000'    lpad( T_Id, 8, 0 )</code>	<input checked="" type="checkbox"/>
Quartier.Gebaeude (OIDMadness_V1.Quartier)	INTERLIS.UUIDOID	<code>uuid("WithoutBraces")</code>	<input type="checkbox"/>
Fluss (OIDMadness_V1.Natur)	OIDMadness_V1.TypeIDShort	<code>'    uuid("WithoutBraces")</code>	<input type="checkbox"/>
See (OIDMadness_V1.Natur)	OIDMadness_V1.TypeID	<code>"    uuid("WithoutBraces")</code>	<input type="checkbox"/>
Wald (OIDMadness_V1.Natur)	INTERLIS.UUIDOID	<code>uuid("WithoutBraces")</code>	<input type="checkbox"/>
BesitzerIn ...	INTERLIS.ANYOID	<code>'    uuid("WithoutBraces")</code>	<input type="checkbox"/>

**Reset the technical sequence used as t\_id value**

T\_Id (Sequence) Value

Use logins from settings (superuser) for this operation

# EXKURS 3 INTERLIS KONTROVERSE

# ARGUMENTE GEGEN INTERLIS

- INTERLIS wird vor allem von Behörden in der Schweiz und deren Auftragnehmer verwendet
- Für andere Projekte wird nach meiner Erfahrung kaum INTERLIS verwendet
- Das ist oft so weil...
  - Die Arbeit mit INTERLIS das schnelle einsehen von Daten schwieriger macht
  - Einen Mehraufwand wenn Daten nach einer Analyse in ein Modell geschrieben werden müssen
  - Wenig Wissen und Übung im Umgang mit INTERLIS vorhanden ist.

# ARGUMENTE FÜR INTERLIS

- Wer eine mittlere bis grossere GIS-Infrastruktur betreibt, braucht:
  - Datenvalidierungen
  - Datenmodelle, UML-Diagramme
  - Werkzeuge, um Daten zu exportieren und importieren
  - Dokumentation der Datenstruktur

**INTERLIS bietet dies alles gratis**

**FEED**

**BACK**

[OPENGIS.CH/DE/KURSBEWERTUNG](https://opengis.ch/de/kursbewertung)



**DAS WARS...**

**Web:** [www.opengis.ch](http://www.opengis.ch)

**Email:** [tobias.heini@opengis.ch](mailto:tobias.heini@opengis.ch)

