

GIV – Interlis2 Tool

Dokument: High Level Design (HLD)
Zusammenfassung: Das Interlis2 Tool vom GIS Verbund Thurgau konvertiert Interlis2-Datendateien von verschiedenen Quell-Modellen in Ziel-Modellen.

Das Tool wird öffentlich als Open Source Projekt zur Verfügung gestellt.

Change-Log:
16.09.2021 (USO) Erstellen der Dokumentation.

1. Allgemein

a. Zweck

Dieses Dokument beinhaltet die Anforderungen, Software-Architektur sowie das Software-Design für das Tool «GIV Interlis2 Tools».

b. Gültigkeit

Dieses Dokument ist für die gesamte Lebensdauer des Projektes gültig und wird laufend erweitert und angepasst.

c. Bezeichnungen und Namespace

Projekt Name:	GIV.Interlis2.Tools
Produkt Name:	GIV Interlis2 Tools
Assembly-Name:	GIV Interlis2 Tools.exe
Namespace:	GIV.Interlis2.Tools.*

Applikations-Titel:	GIV-TG – Interlis2 Tool
Deployment:	EXE (portable)
Installationspfad:	-
Startmenu:	-
Konfigurationsdatei:	-

d. Abkürzungen

DSS	Datenstruktur Siedlungsentwässerung
MEL	Meliorationen
GEP	Generelle Entwässerungsplanung

2. Anforderungen (Übersicht)

In diesem Abschnitt werden die essenziellen Anforderungen dokumentiert.

ID:	Bezeichnung:	Datum/Owner:
RE001	Konvertieren vom Modell DSS in das Modell GEP	27.01.2021 TK Werke
RE002	Konvertieren vom Modell DSS in das Modell MEL	27.01.2021 TK Werke
RE003	Aus dem Datenbestand DSS den Datenbestand MEL extrahieren	27.01.2021 TK Werke

3. FU001: Modelle Konvertierungen

Diese Funktion wird für die Konvertierung von einem Quell-Modell zu einem Ziel-Modell verwendet.

a. Abdeckung Anforderungen

RE001: Konvertieren vom Modell DSS in das Modell GEP

RE002: Konvertieren vom Modell DSS in das Modell MEL

b. Grundsätzliche Funktionsweise

Die Konvertierung verwendet lediglich die Daten Dateien (XTF) und ignoriert die entsprechenden Modelle (ILI).

Es werden keine Prüfungen bezüglich Korrektheit oder Modell-Konformität gemacht.

Die Konvertierung basiert auf einer einfachen Text Ersetzung (String.Repalce)

c. Model-Node

Im XTF in der «HEADERSECTION» ist die Angabe des «MODEL» gespeichert.

```
<HEADERSECTION VERSION="2.3" SENDER="GEOBOX AG">
  <MODELS>
    <MODEL NAME="DSS_2020_LV95" VERSION="25.06.2021" URI="http://www.vsa.ch/models" />
  </MODELS>
</HEADERSECTION>
```

Diese Angaben wird überschrieben mit den Angaben des Ziel-Modelles, wenn der Model-Node für das Quell-Modell vorhanden ist.

Die Angaben für das Ziel-Modell sind im jeweiligen Controller hart codiert.

d. Daten Nodes

Bei der Konvertierung werden lediglich die XML-Tags von DSS auf GEP oder MEL umgeschrieben.

Konkret wird der Namespace des XML-Tags umgeschrieben mit einer Textersetzung.

Beispiel:

String.Replace(«DSS_2020_LV95», «DSS_2020_LV95_GEP»)

4. FU002: Meliorationsdaten extrahieren

Diese Funktion filtert Datenbestände aus einer Quell-Datendatei und speichert die gefilterten Daten in eine neue Datendatei (Ziel-Datendatei).

a. Abdeckung Anforderungen

RE003: Aus dem Datenbestand DSS den Datenbestand MEL extrahieren

b. Grundsätzliche Funktionsweise

Die Auftrennung erfolgt mittels Filterangaben (Funktionen-Melioration) und den verbundenen Elementen.

Die Verbundenen Elementen werden mit den Referenzattributen entsprechend gesucht (TID/OID Werte).

Es werden keine Prüfungen (Interlis-Checks) gemacht, sondern auf der Basis der XTF-Dateien die Datensätze extrahiert und gespeichert.

c. Referenzen

Die Referenzen werden nicht aus dem Interlis-Modell (ILI) gelesen, sondern sind im Kontroller fest eingestellt.

d. Irrelevante Objektklassen

Datensätze der Irrelevanten Objektklassen werden ohne Prüfung entfernt, respektive nicht übertragen.

e. Filterklassen und Filterattribute

Die folgenden Klassen und Attribute werden als Filter verwendet:

Klasse:

Kanal

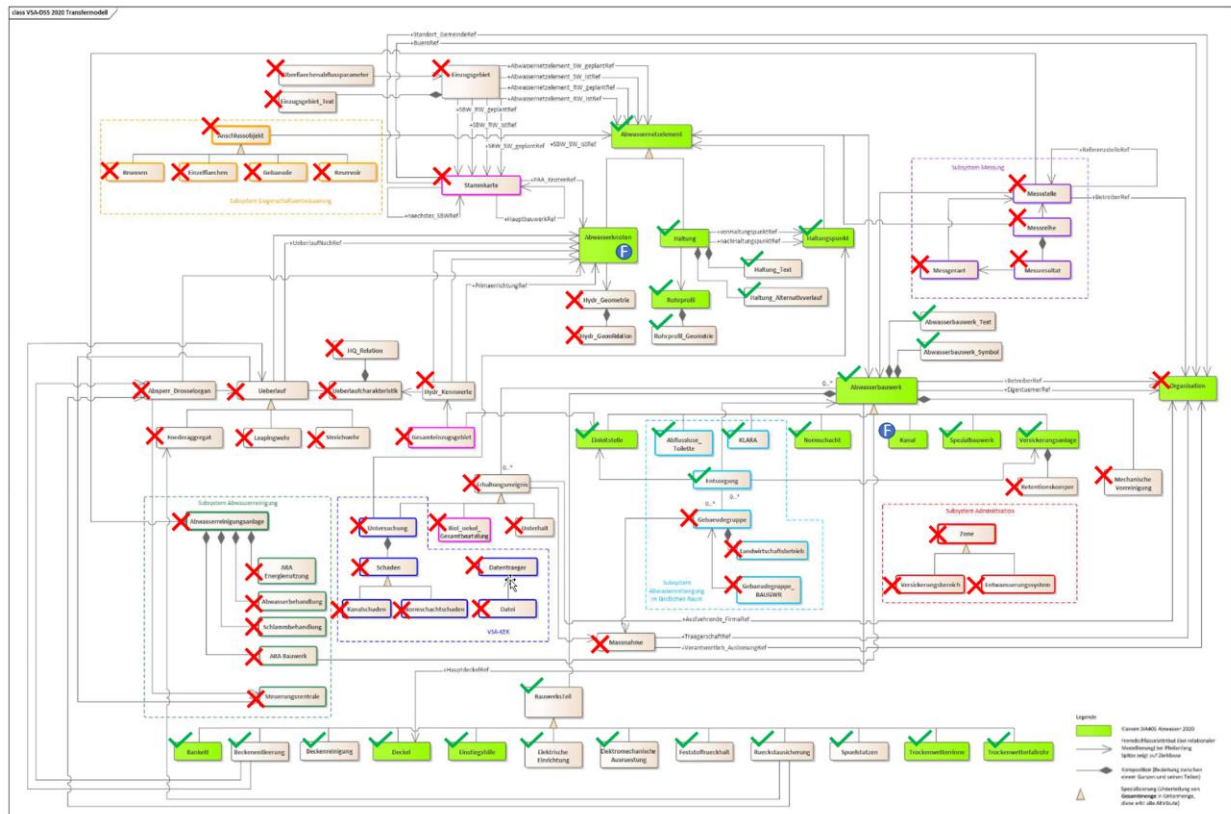
Abwasserknoten




Attribut:

FunktionMelioration

Funktion_Knoten_Melioration

f. Objektklassenübersicht



Symbol:	Bezeichnung:	Beschreibung:
	Filterklasse	Klassen bei denen die Filter angewendet werden und die entsprechenden Datensätzen werden übertragen.
	Referenzierte Klasse	Diese Klassen beinhaltet Daten, die in Abhängigkeit (Referenz) zu den Filterklassen beinhalten. Hier werden die entsprechenden Referenzierten Datensätze übertragen.
	Irrelevante Klasse	Daten in diesen Klassen werden nicht übertragen.

5. SAD – Kommandozeilen Argumente

a. Argumente

Argument:	Kurz:	Beschreibung:
--type	-t	Angabe der Anzuwendenden Funktion (siehe detaillierte Beschreibung)
--input	-i	Pfad und Dateiname der Eingabe (Input) Datei. In der Regel das XTF mit den Quelldaten.
--output	-o	Pfad und Dateiname der Ausgabe (Output) Datei. In der Regel das XTF für die Speicherung der Konvertierten Daten.
--log	-log	Pfad und Dateiname für die Protokoll Datei.

b. Argument «--type»

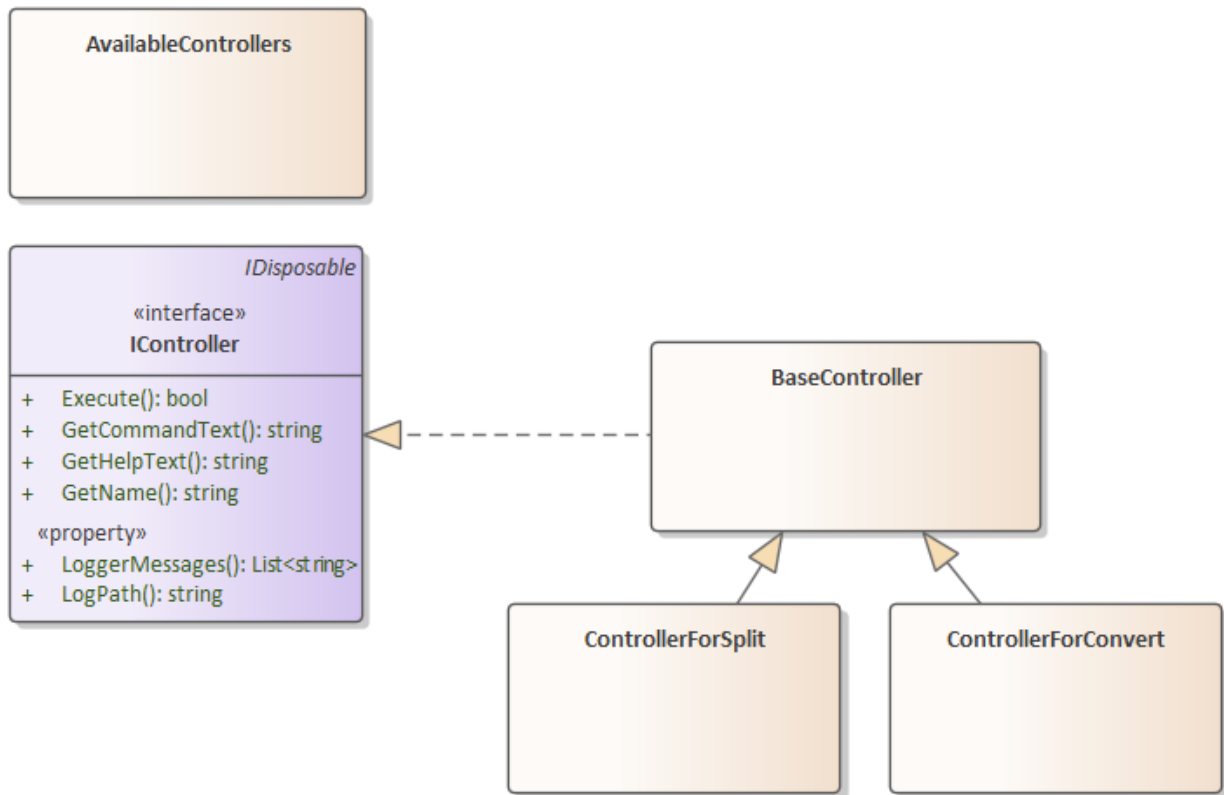
Das Argument «--type» wird für die Angabe der Funktion (Konvertierung oder Trennung) verwendet.

Für die Benutzenden muss dieser Wert in der Benutzeranleitung bekannt gemacht werden.

Der Wert wird in der Konkreten Implementation des Kontrollers als Konstante definiert und in den Verfügbaren Kontrollers (AvailableControllers) eingetragen.

6. SAD – Controllers

Die «Controllers» definieren die verschiedenen Aufgaben innerhalb der Applikation.



a. AvailableControllers

Jeder konkrete Implementation eines «Controllers» (:IController) muss in der Liste eingetragen werden.

Dies mit dem Funktionsname (FUNCTIONTYPE) der Gleichzeitig als Kommandozeilen Argument für den Aufruf verwendet wird.

b. BaseController : IController

Stellt die Funktion über allen Controller sicher.

c. Konkrete Implementationen

Die Konkreten Implementationen leiten vom BaseController ab und Implementieren so das Interface IController.

Für die Controller ist die Vorgabe, dass die Konstante «FUNCTIONTYPE» definiert wird, mit einem Eindeutigen Wert (UNIQUE) für die Erkennung des Controllers per Kommandozeilen Argument (`--type`).