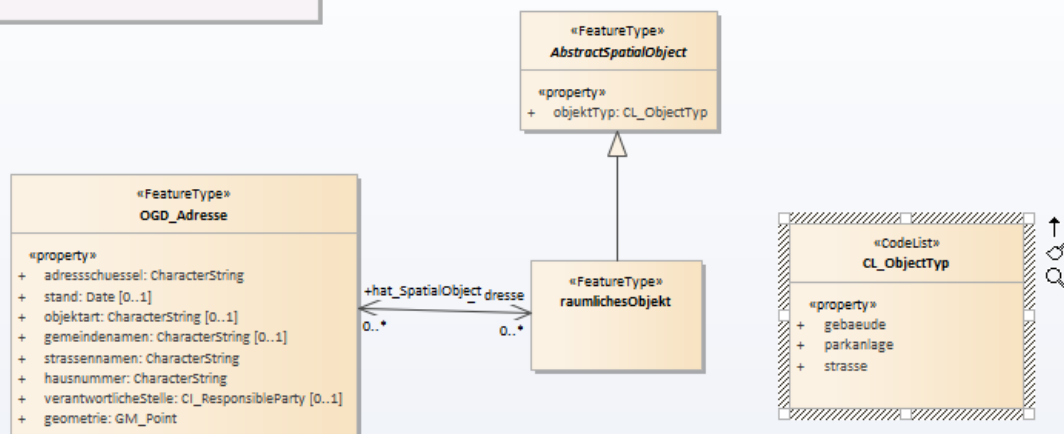




Erstellen eines Applikationsschemas im Enterprise Architect

Applikationsschema: OGD Daten Tirol
Erstellt von: Roland Grillmayer
Version 1.0



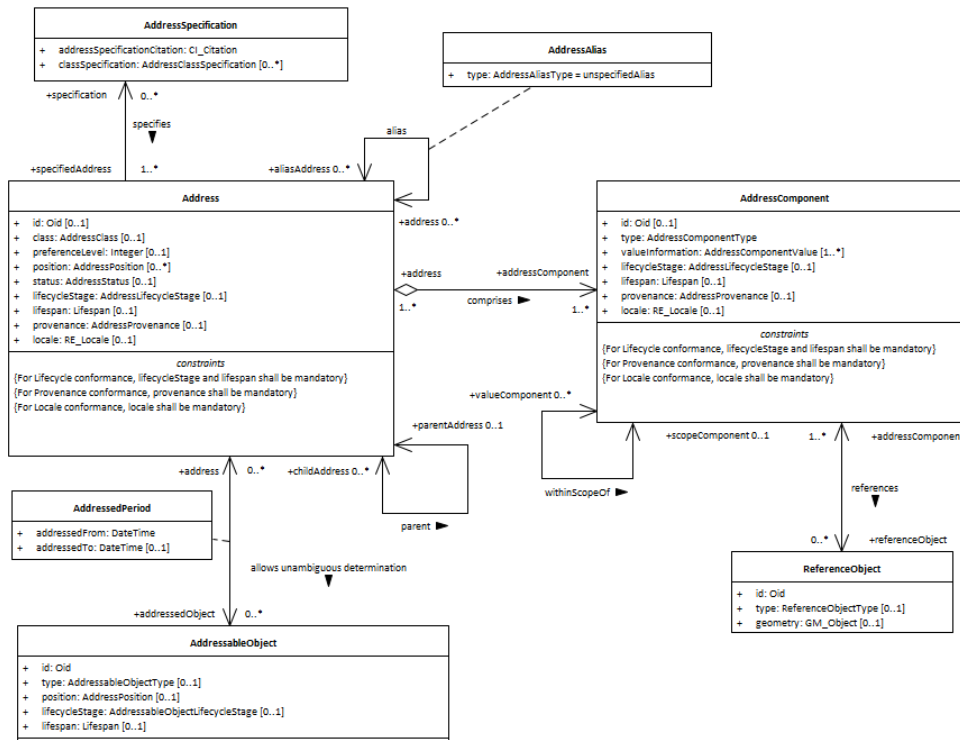
OGD Adressen Tirol

Erstellen eines UML Diagramm für das Applikationsschema OGD Adressen Tirol

ISO 19160-1:2015

Preview

Addressing -- Part 1: Conceptual model



1.) Allgemeine Informationen - Erstellen eines UML Diagramms im Enterprise Architect für den Datensatz OGD Adressen Tirol

- Welche Informationen sind im Ausgangsdatsatz verfügbar?
 - https://www.data.gv.at/katalog/dataset/land-tirol_adressentirol
 - Keine Informationen zu den Attributausprägungen
 - https://gis.tirol.gv.at/arcgis/services/Service_Public/OGD/MapServer/WFSServer?request=DescribeFeatureType&service=WFS

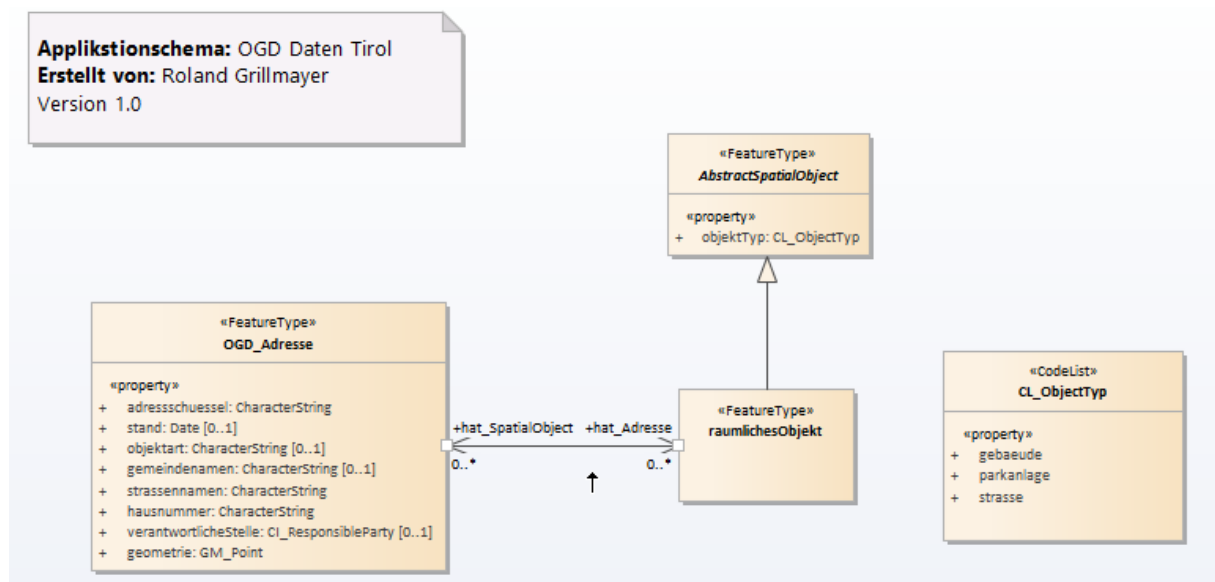
```

<xsd:complexType name="GEO_AdressenFeatureType">
  <xsd:complexContent>
    <xsd:extension base="gml:AbstractFeatureType">
      <xsd:sequence>
        <xsd:element minOccurs="0" maxOccurs="1" name="OBJEKT" nillable="true" type="xsd:string"/>
        <xsd:element minOccurs="0" maxOccurs="1" name="GEMNR" nillable="true" type="xsd:int"/>
        <xsd:element minOccurs="0" maxOccurs="1" name="PLZ" nillable="true" type="xsd:int"/>
        <xsd:element minOccurs="0" maxOccurs="1" name="GEMNAME" nillable="true" type="xsd:string"/>
        <xsd:element minOccurs="0" maxOccurs="1" name="SNAME" nillable="true" type="xsd:string"/>
        <xsd:element minOccurs="0" maxOccurs="1" name="HNR" nillable="true" type="xsd:string"/>
        <xsd:element minOccurs="0" maxOccurs="1" name="ADRKEY" nillable="true" type="xsd:string"/>
        <xsd:element minOccurs="0" maxOccurs="1" name="SHAPE" nillable="true" type="gml:PointPropertyType"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

```

Modellierung eines FeatureTypes der eine 0..* Assoziation zu einem räumlichen Objekt aufweist. Das räumliche Objekt wird durch die Codeliste CL_Objecttyp näher spezifiziert in (gebäude | parkanlage | strasse...)

Das fertiggestellte Applikationsschema soll folgendermaßen ausschauen



Das fertiggestellte EA-Projekt ist unter:

\\Desktop\INSPIRE\AD\Adressen_fertiggestellt.eap verfügbar

2.) Installation und Konfiguration des Enterprise Architect im Linux

Unter Nutzung von Wine ist es möglich den EA unter Debian-Linux Distributionen zu betreiben.
Infos dazu unter:

https://sparxsystems.com/enterprise_architect_user_guide/14.0/product_information/enterprise_architect_linux.html

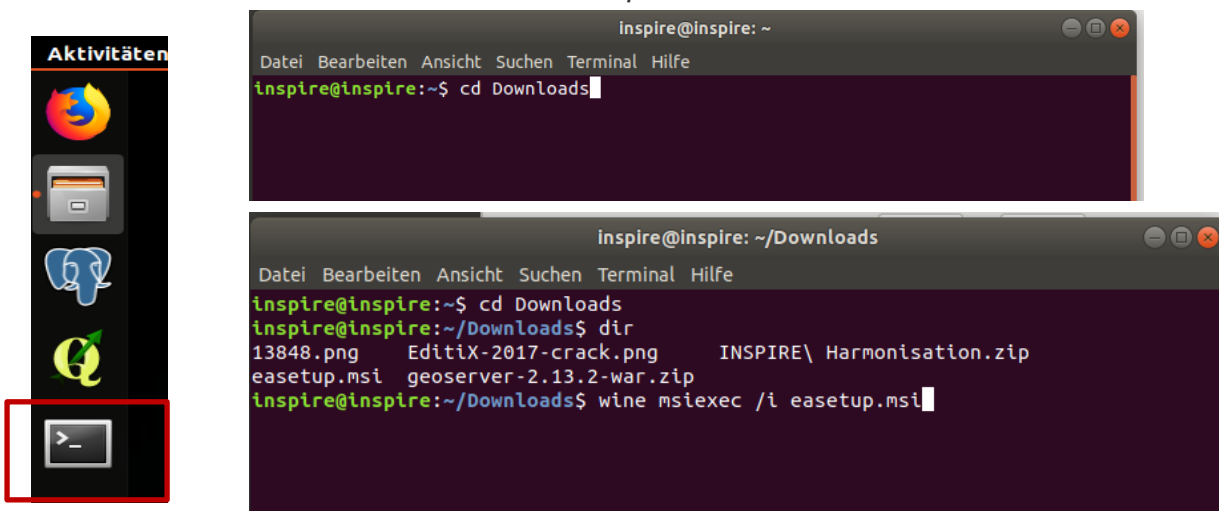
Am bereitgestellten Image wurden bereits die benötigten Softwarekomponenten installiert, so dass der EA direkt wie folgt



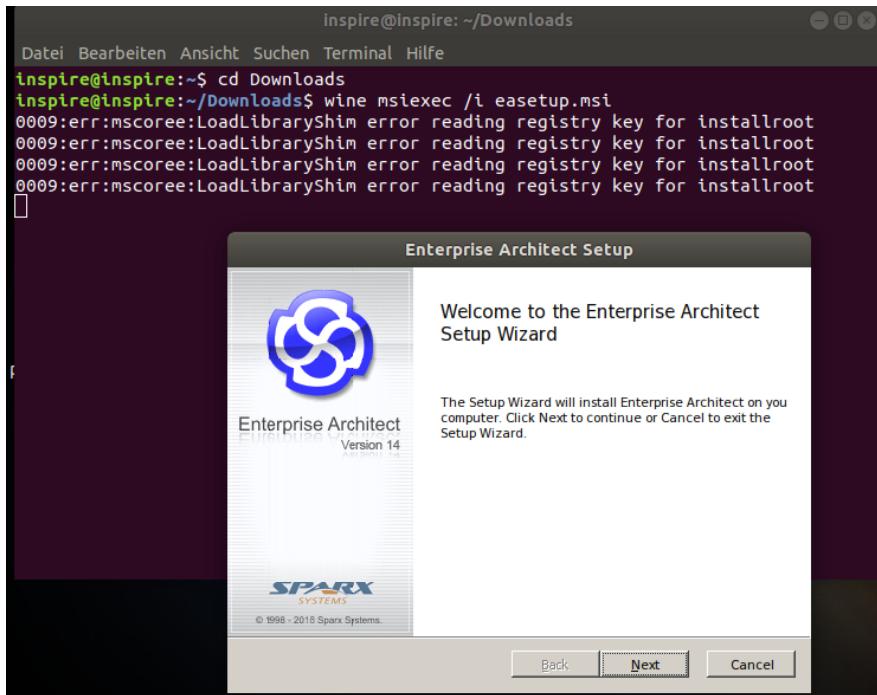
installiert werden kann. Bei der zur Verfügung gestellten Datei handelt es sich um ein 30 Tage Testversion.

Installation der Demo Version des EA (30 Tage Laufzeit)

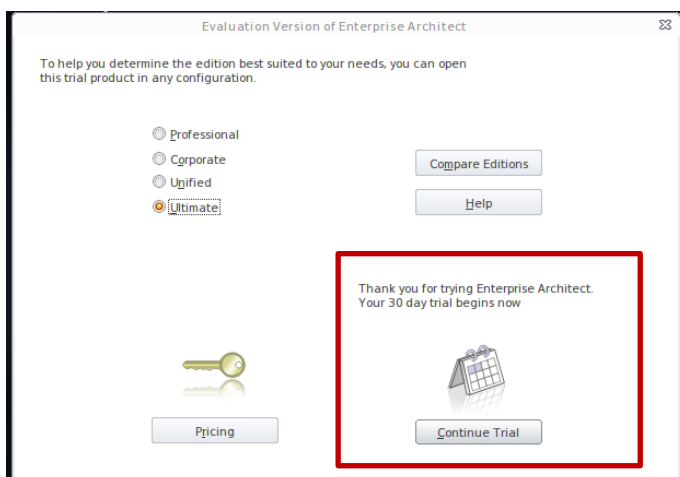
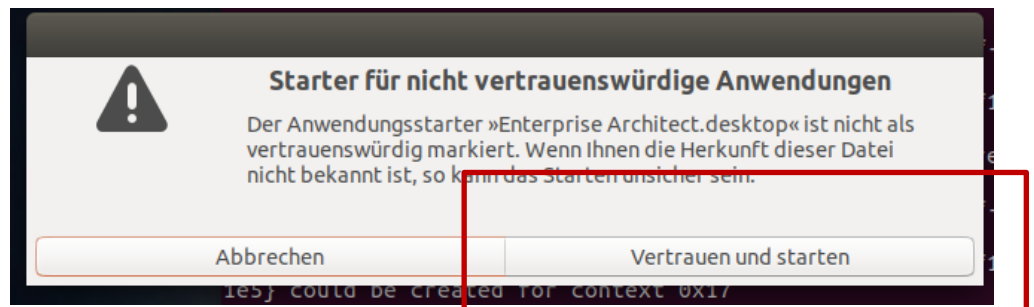
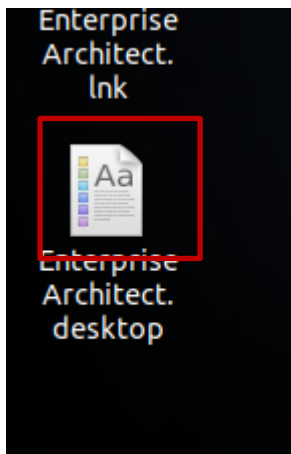
Terminal Fenster öffnen, zum Verzeichnis Download (cd Downloads) navigieren und folgendes Kommando ausführen: `wine msiexec /i easetup.msi`



Die auf der nächsten Folie angeführten Fehlermeldungen ignorieren → Installation des EA startet → dem Wizard folgen und Installation durchführen

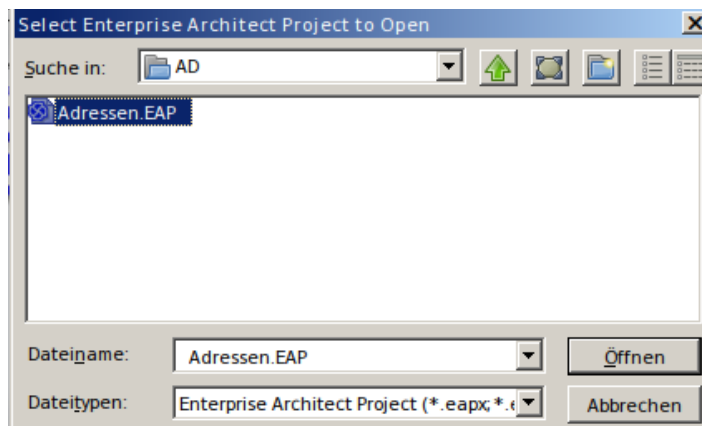
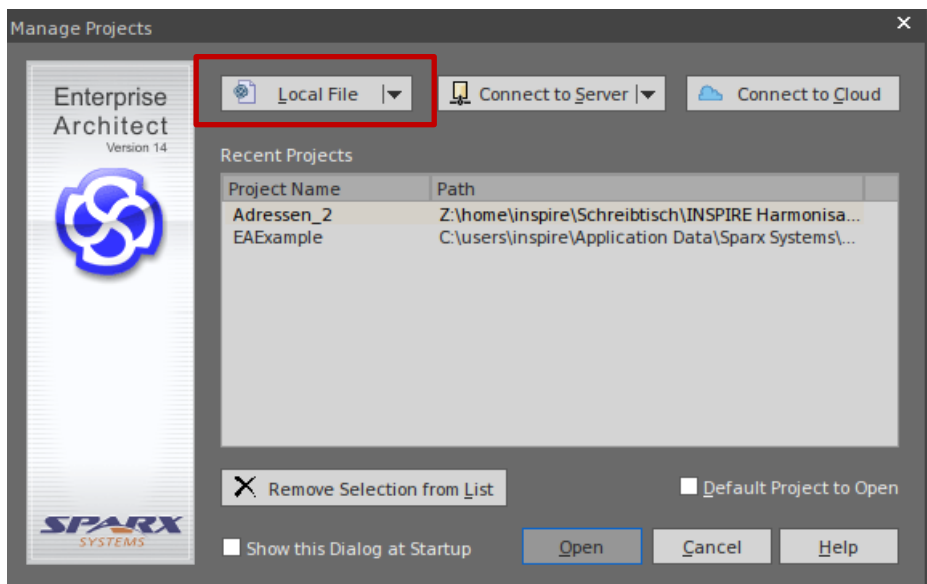
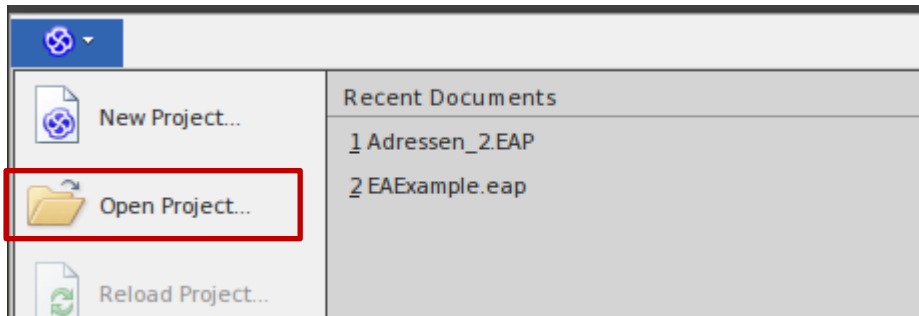


EA starten -30 Tage Testversion aktivieren

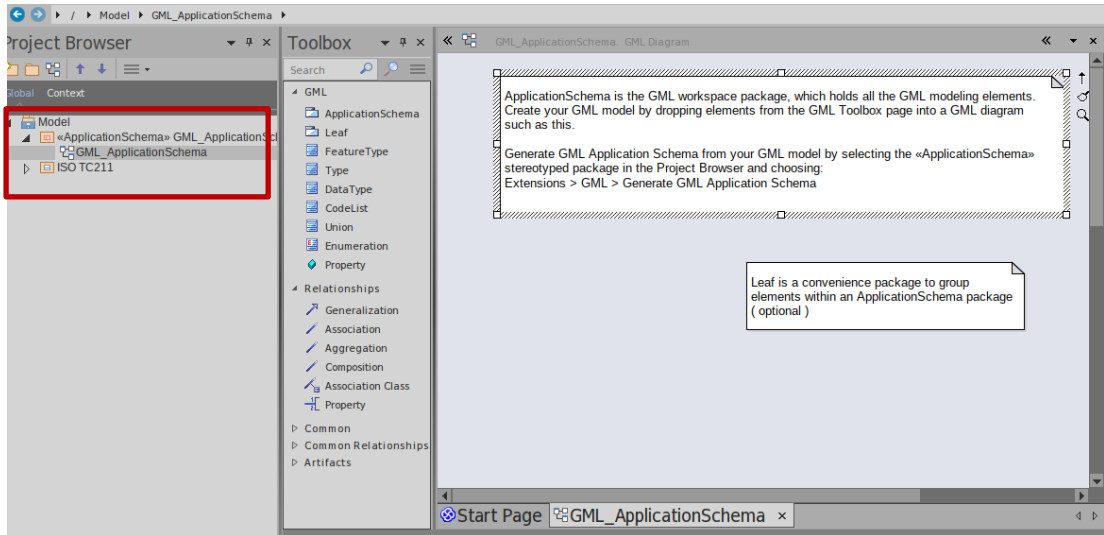


Öffnen des GML-Templates und Konfiguration des EA

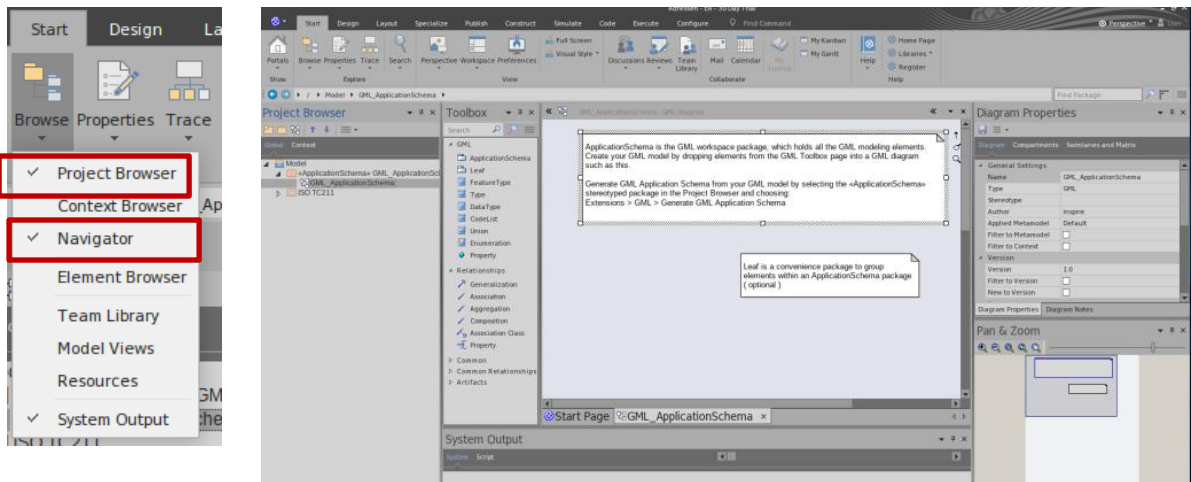
- Öffnen der EA und des Projektes Adressen.eap verfügbar unter [/Desktop/INSPIRE/AD/Adressen.eap](#)



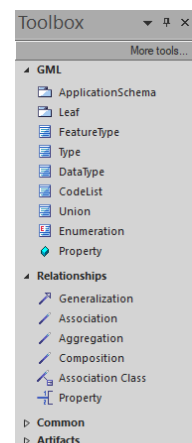
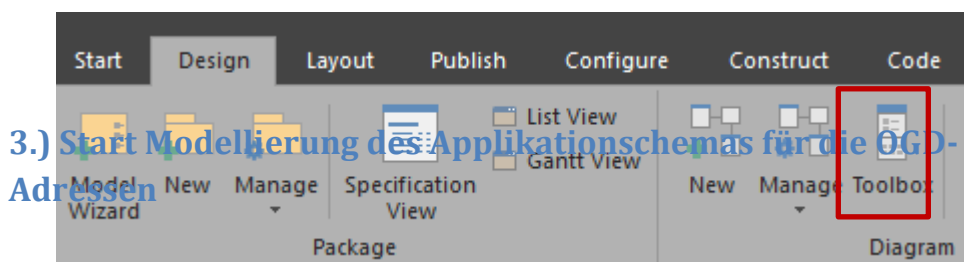
Vorlage für das Modellieren von GML Applikationschema sollte jetzt verfügbar sein. Im Project Browser ist ein leeres Tabellenblatt „GML_ApplicationSchema“ sowie die ISO Datentypen verfügbar (siehe Abbildung unten). Anklicken des GML_ApplicationSchema Datenblattes zum Öffnen der Datenblattansicht.



Allfällige Konfiguration des EA falls Projektbrowser oder andere benötigten Windows nicht zur Verfügung stehen → Benötigte Fenster wie in der Abbildung einblenden

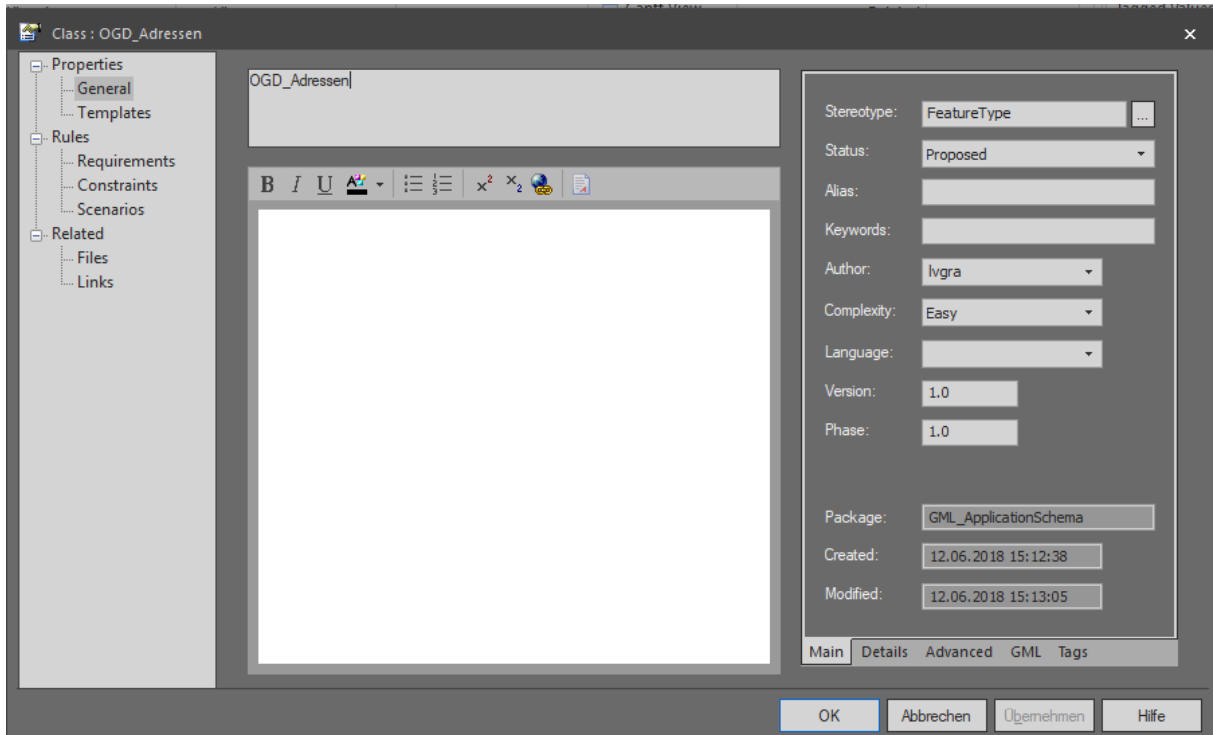


Falls die Toolbox für die GML spezifischen Elemente nicht verfügbar ist – diese über Design → Toolbox aktivieren. Diese stellt alle für die GML Modellierung benötigten UML Konstrukte zur Verfügung



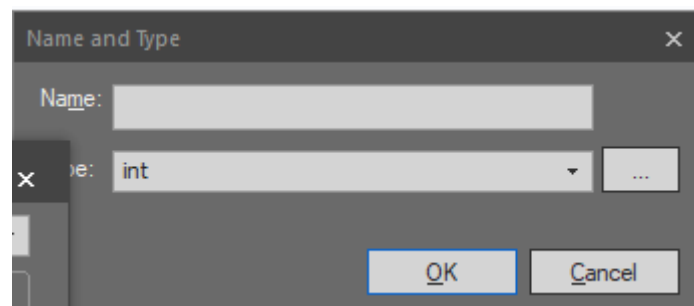
1.) Erstellen des Feature Types „OGD_Adressen“ – Drag and Drop aus der Toolbox

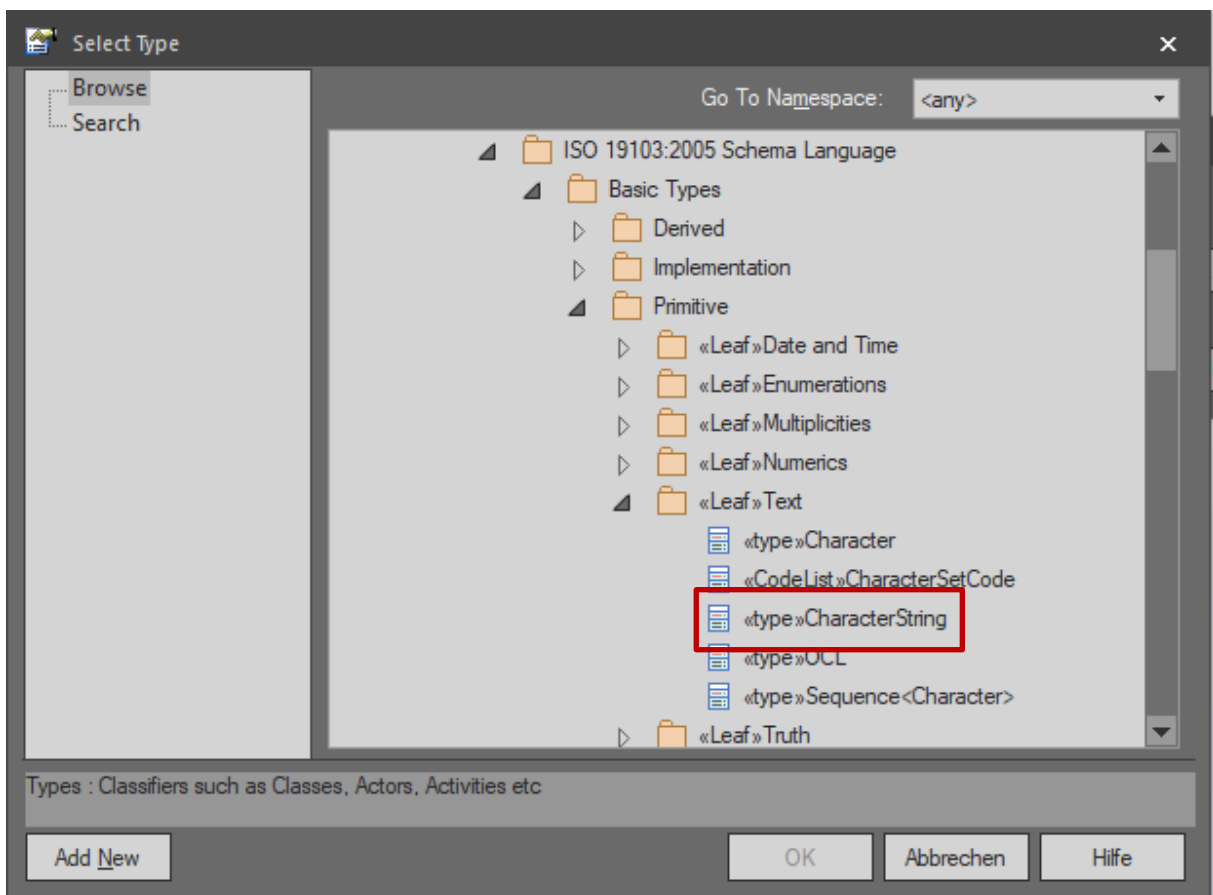
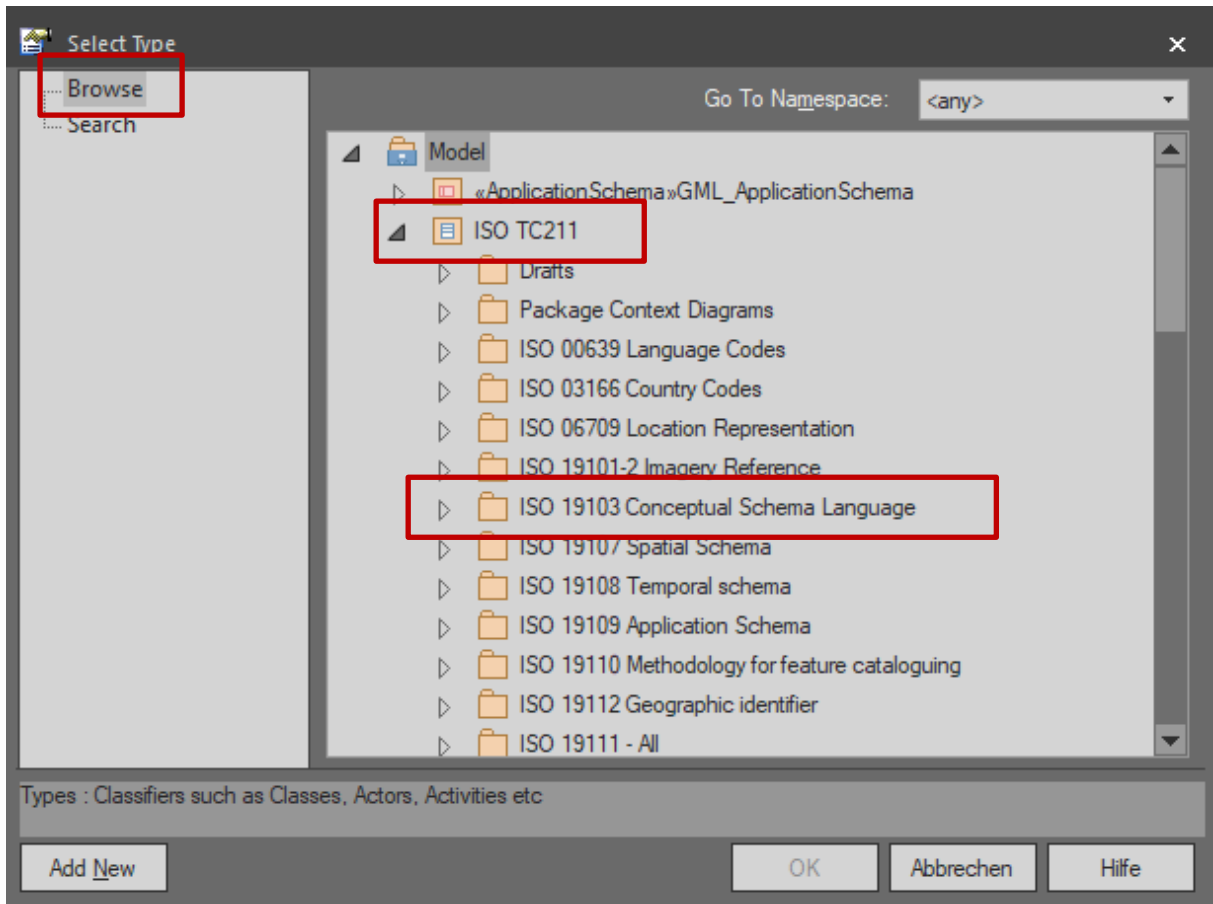
Modifizieren des FeatureTypeName



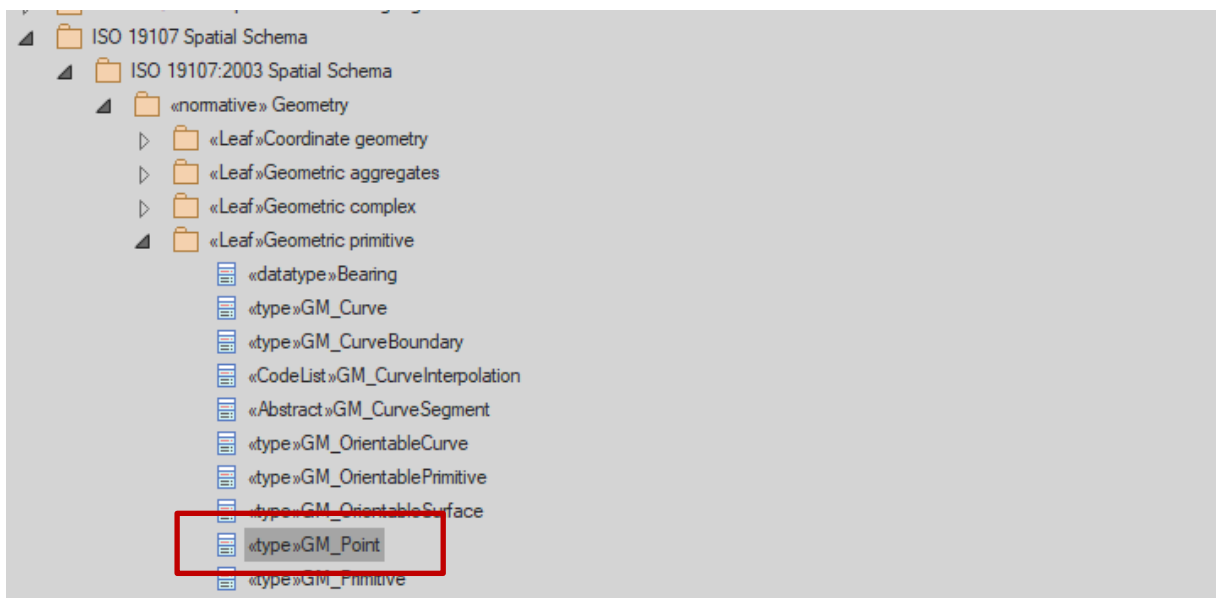
2.) Modellieren der Feature Properties

- Modellieren der Object_ID → Drag & Drop Property von der Toolbox
 - Property Name: adressschluessel
 - Datatype: ISO 19103:CharaterString → erreichbar durch folgenden Workflow

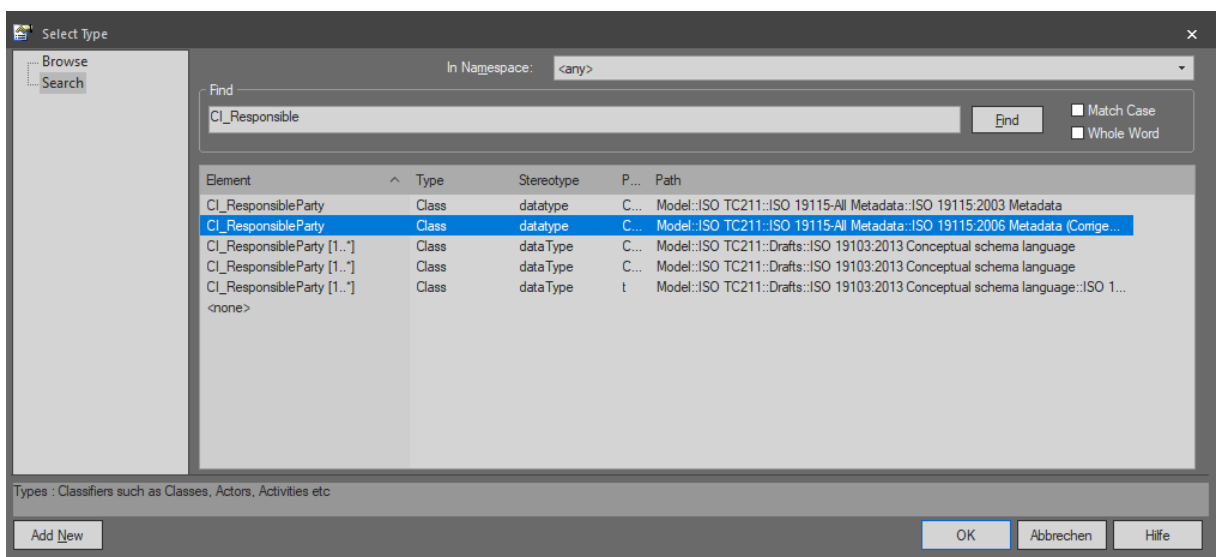




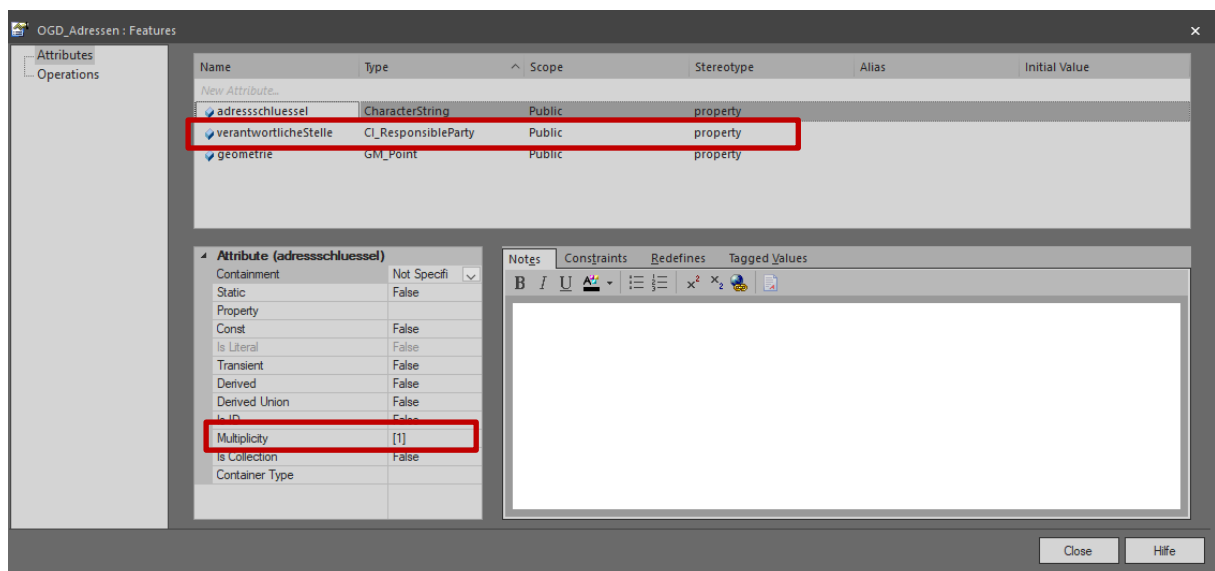
- Modellieren der anderen FeatureProperties der Feature Types OGD_Adressen auf gleiche Art und Weise
 - gemeindenummer: CharacterString (0...1)
 - gemeindenamen: CharacterString (0...1)
 - hausnummer: CharacterString
 - strassenname: CharacterString
 - objektart: CharacterString (0...1)
 - postleitzahl: CharacterString (0...1)
- Modellieren der Punktgeometrie
 - geometrie:ISO 19107 - <<Leaf>> GeometricPrimitiv <<type>>GM_Point



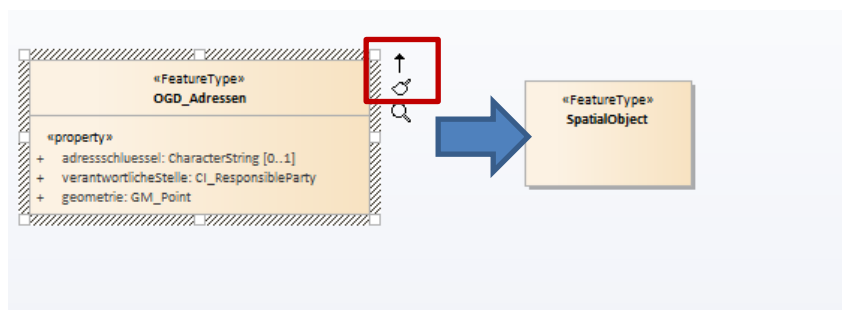
- Modellierung der verantwortlichenStelle unter Nutzung eines komplexen Datentyps der ISO 19155:2006 – Metadaten
 - verantwortlicheStelle: CI_ResponsibleParty (0...1)

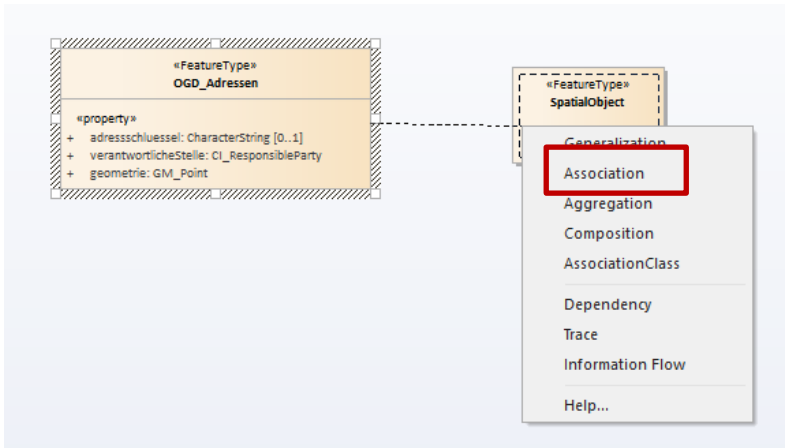


- Modellierung des Aktualitätsdatums
 - Stand: Date (0...1)
- Definition der Kardinalitäten der FeatureProperties
 - Doppelklick auf den FeatureType
 - Strg und Pfeiltasten kann die Reihenfolge modifiziert werden



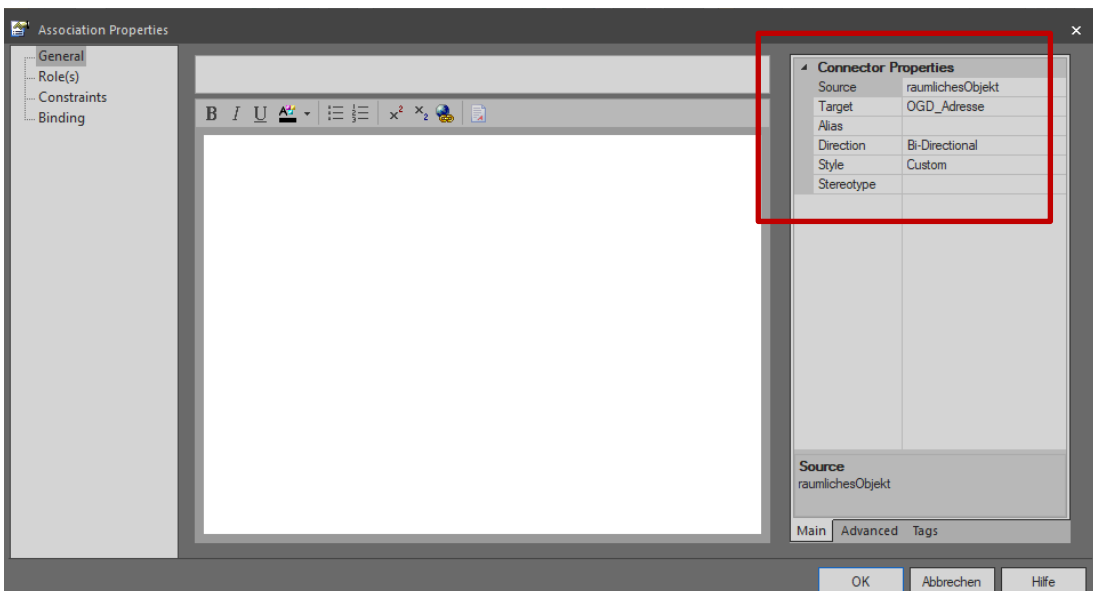
- Modellieren des FeatureType SpatialObject
- Etablierung einer Assoziation zwischen den beiden FeatureTypes OGD_Adressen und SpatialObject



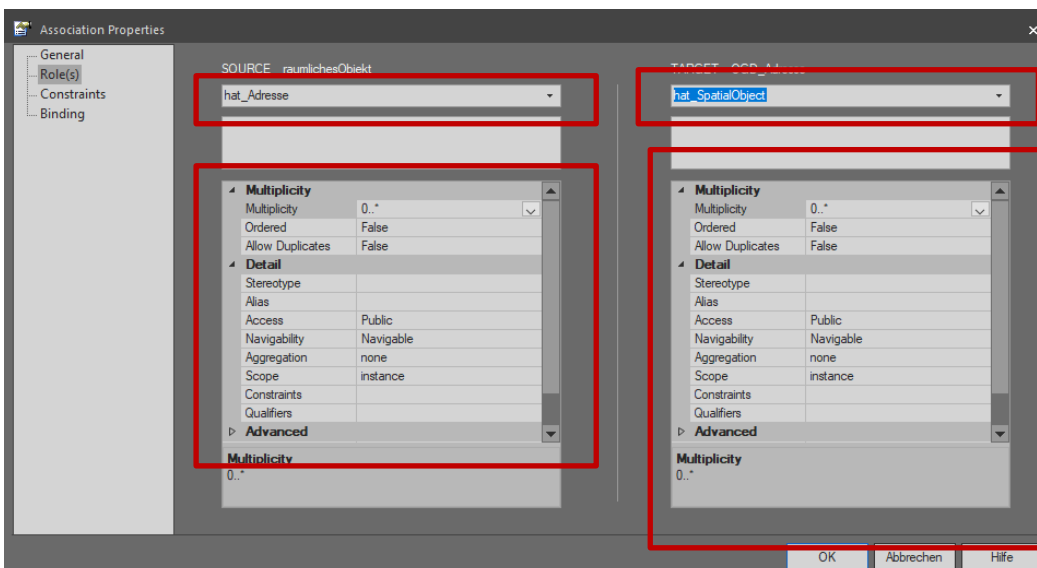


- Doppelklick auf die Assoziation und Definition der Eigenschaften

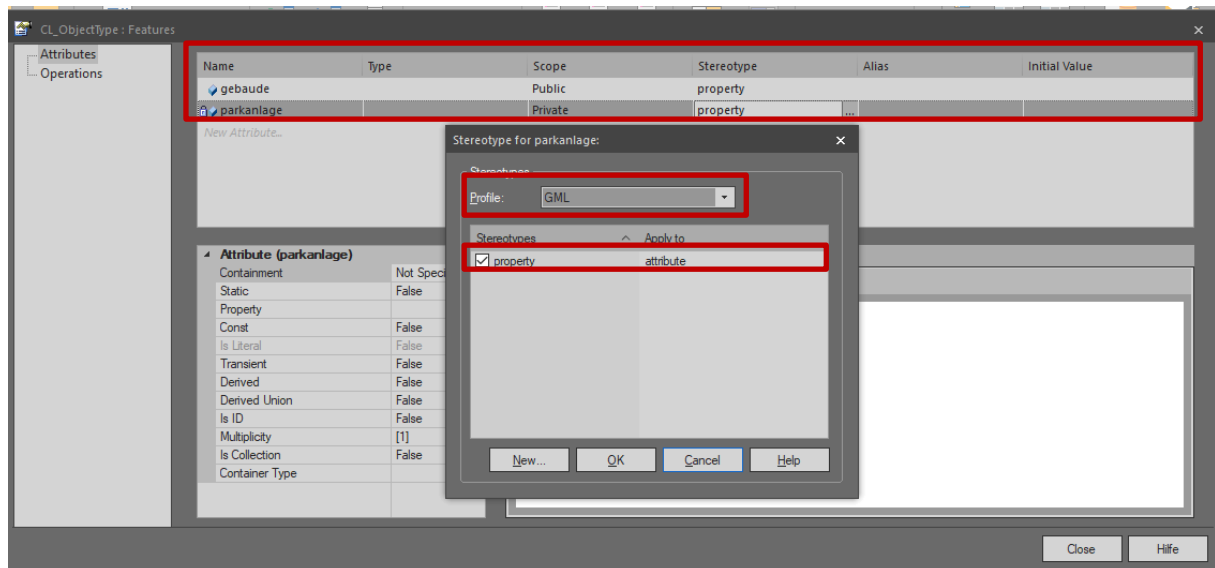
General



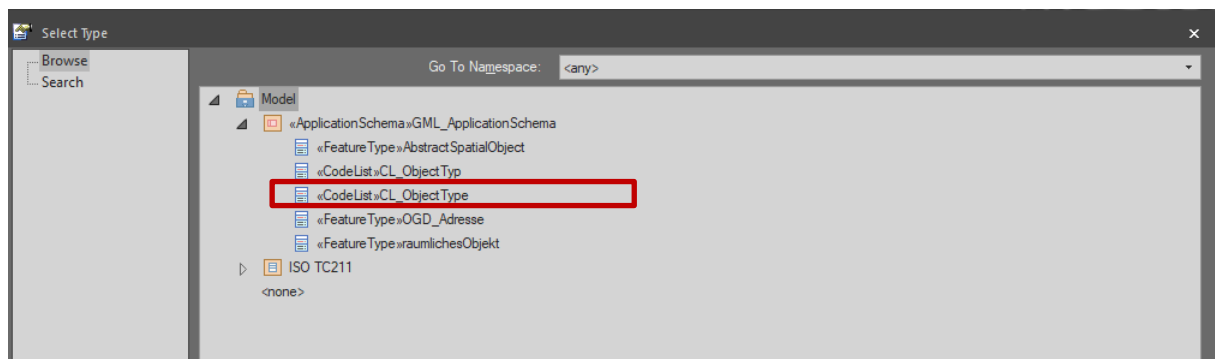
Role(s)



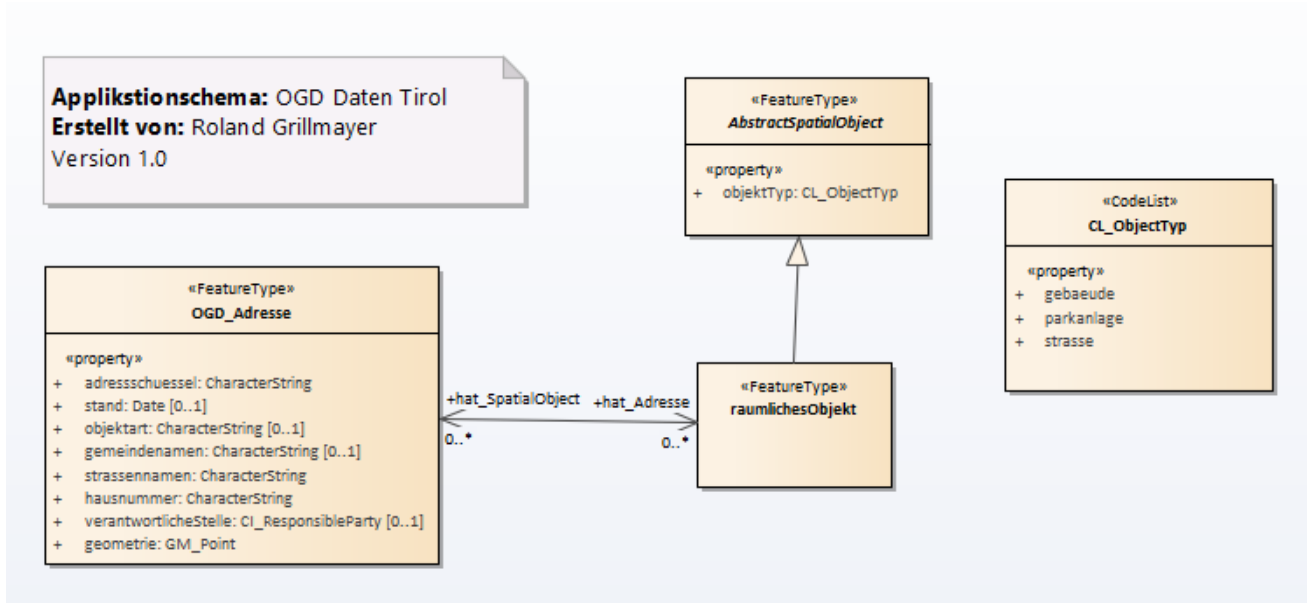
- Erstellen der Codeliste CL_ObjectType
- Codelisten Values werden mit normalen FeatureProperties die keinen Datentypen besitzen modelliert



- Definieren des abstrakten FeatureType AbstractSpatialObject mit der FeatureProperty objectType welche die gerade definierte Codeliste „CL_ObjectType“ nutzt.



Fertiggestellte UML Diagramm aus dem das GML-Schema abgeleitet wird

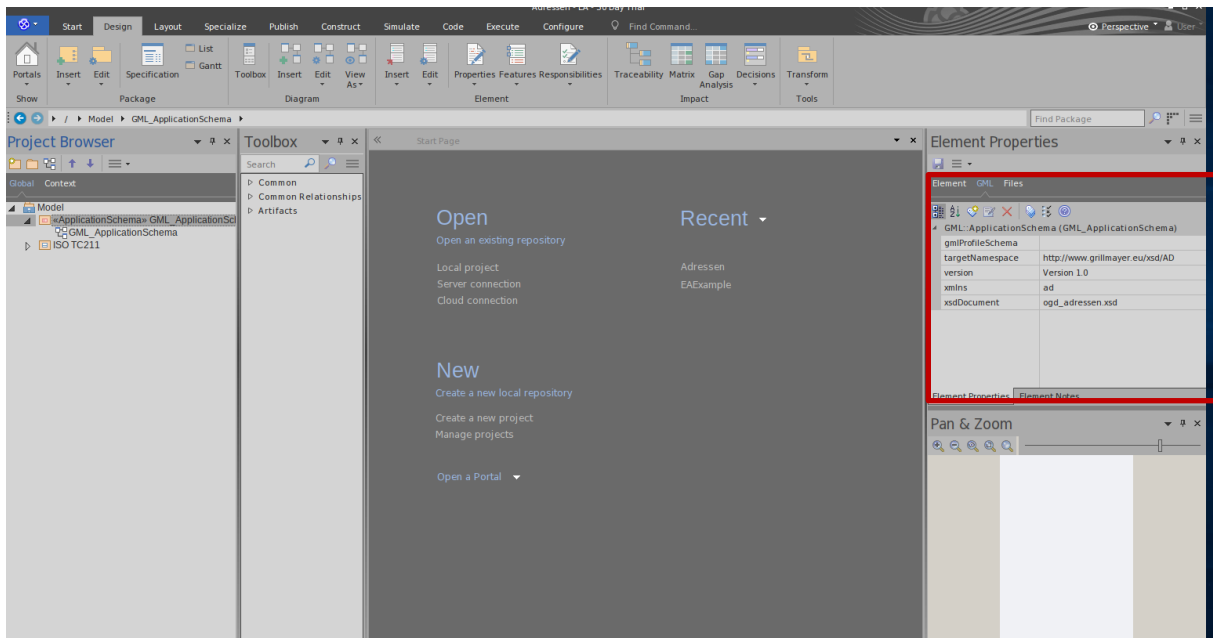


Ableiten des GML Schemas

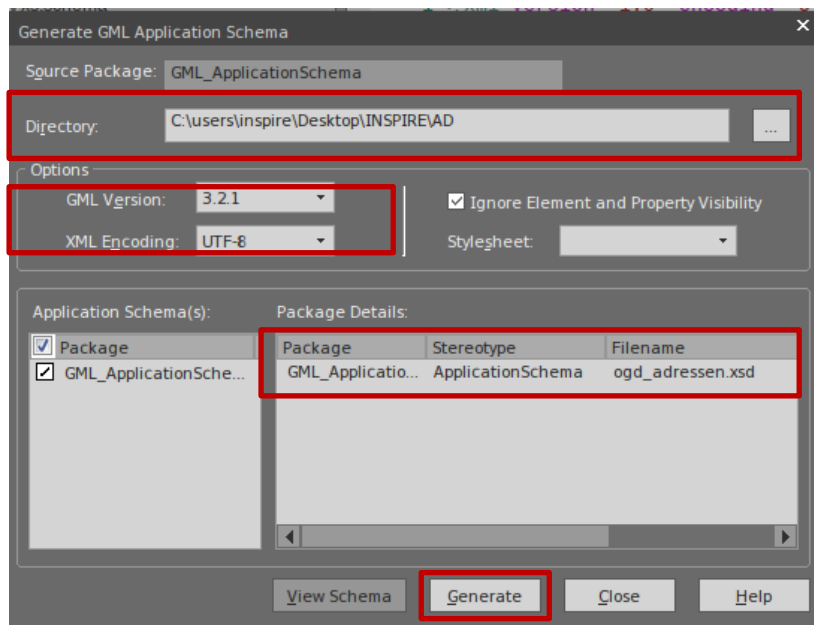
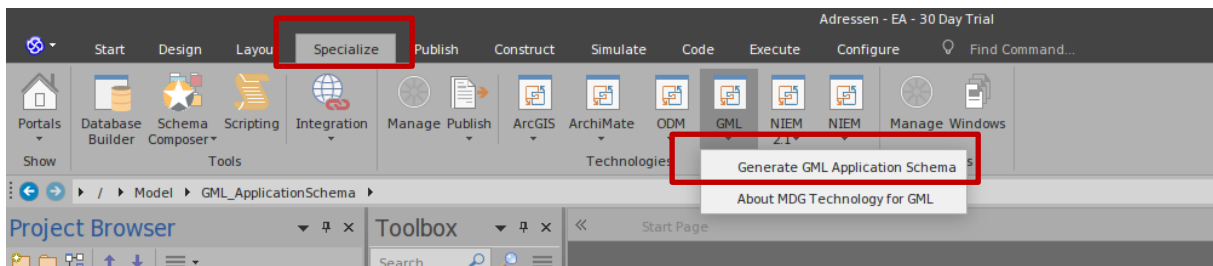
Konfigurieren der GML_Application_Schema Eigenschaften wie folgt

Rechter Mouse Klick auf <<ApplicationSchema>> GML.... Im Project Browser

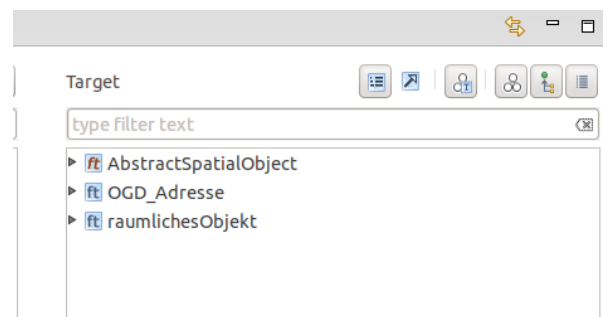
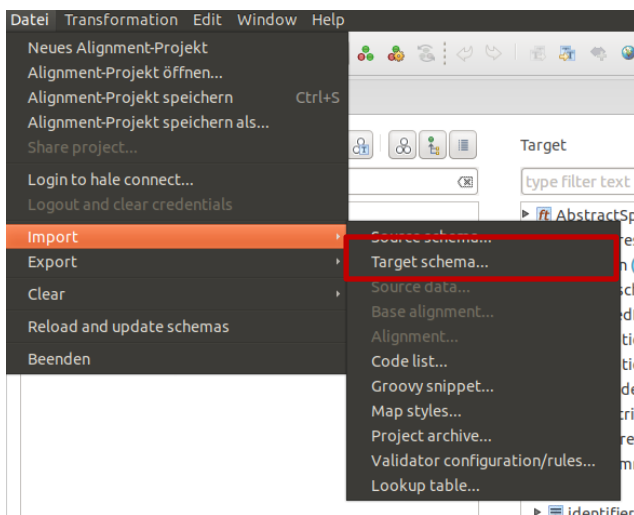
Im Element Properties Fenster (Rechts) Angaben zum targetNamespace | Version | xmlns | und xsdDocument ausfüllen



Ableiten des Schemas → Specialize → GML → Generate GML Application Schema




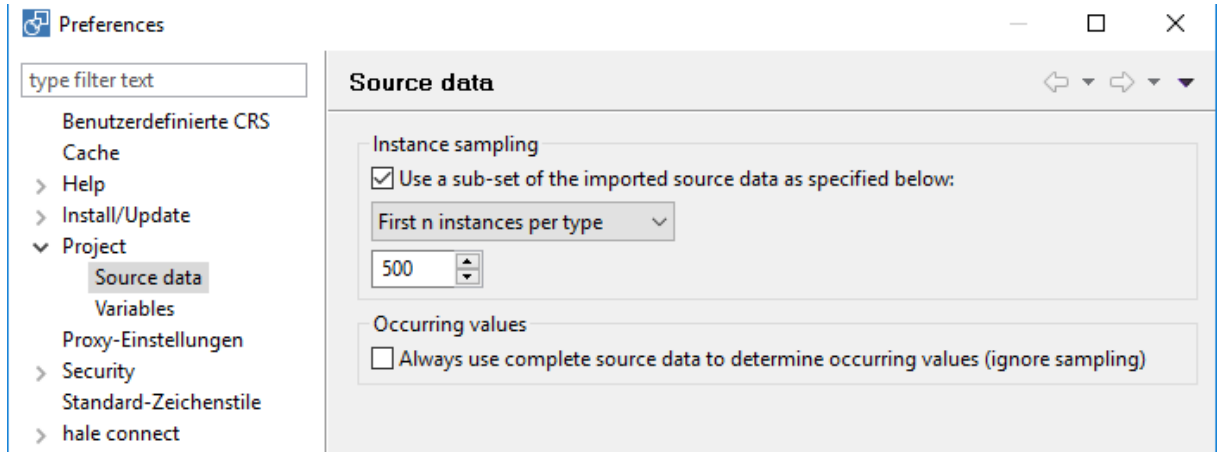
- Überprüfen des Schemas im HALE Studio
Laden des Schemas im HALE Studio – Überprüfen der FeatureTypes , Feature Properties, Associationen, Kardinalitäten.....



4.) Datenharmonisierung in der OGD_Adressen Tirol HALE Studio

2.1.) Projekt einrichten

- Instance Sampling auf 500 konfigurieren 



Datenbankverbindung

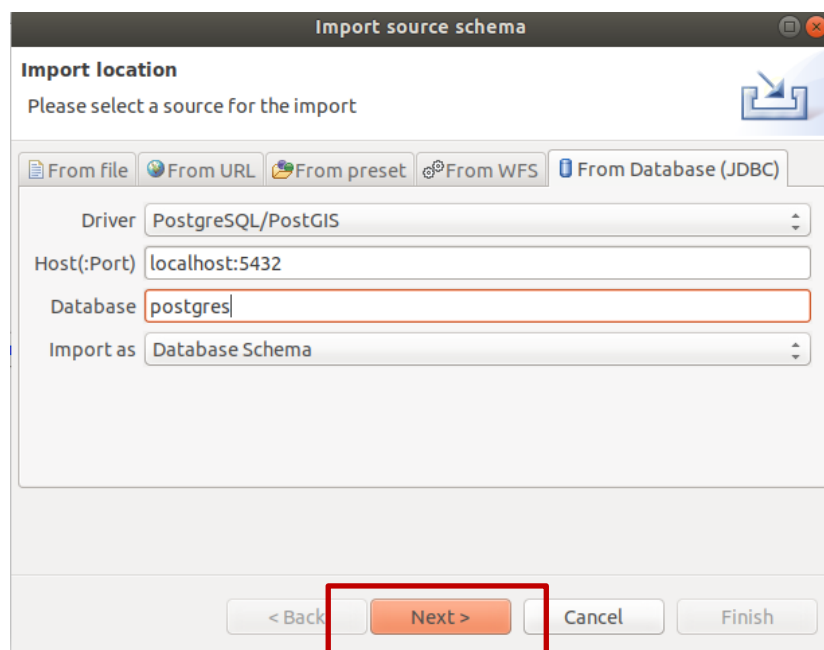
=====

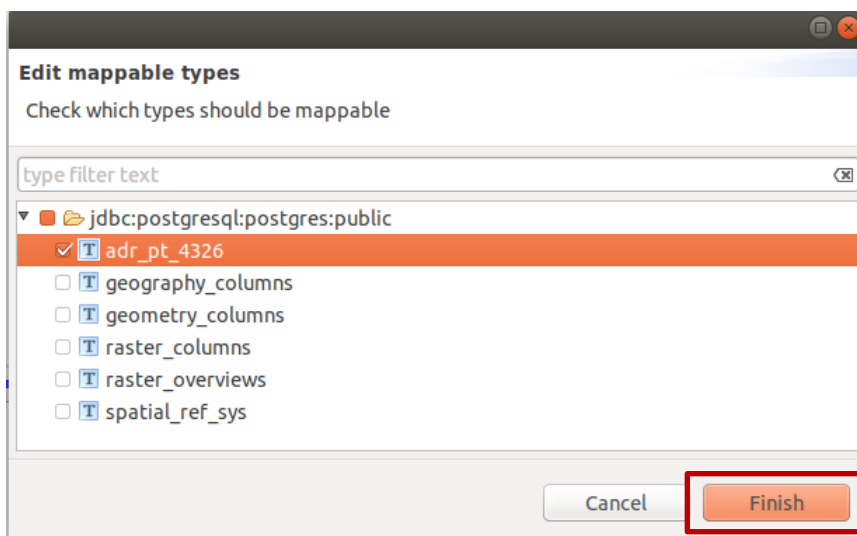
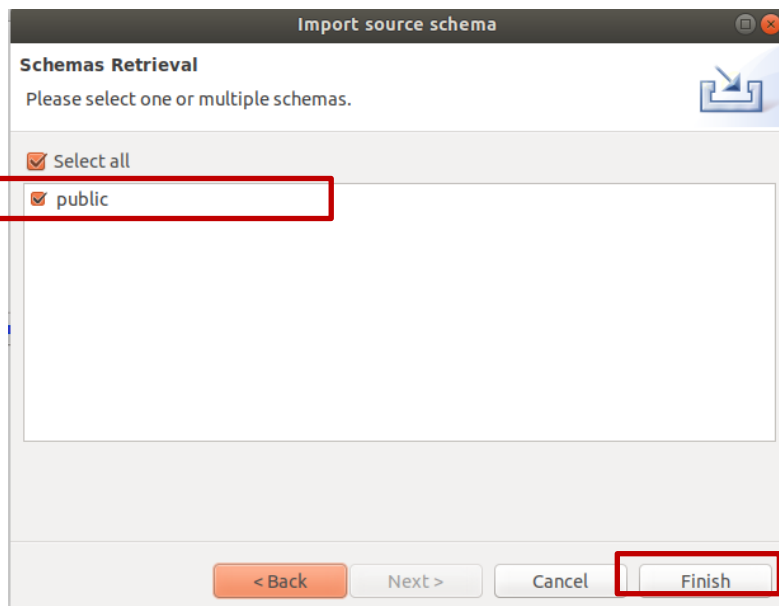
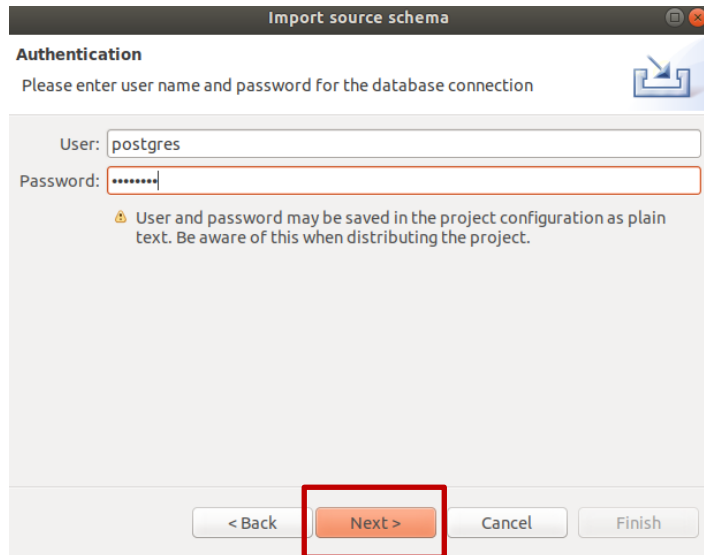
Host: localhost:5432

User: postgres

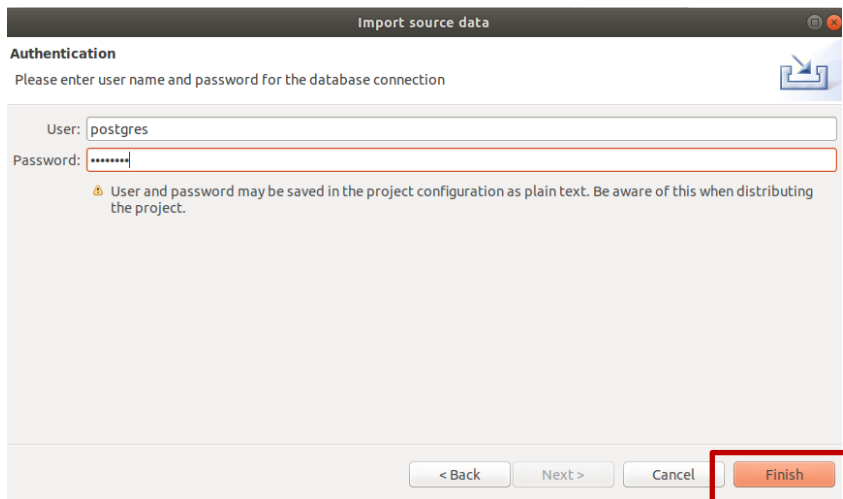
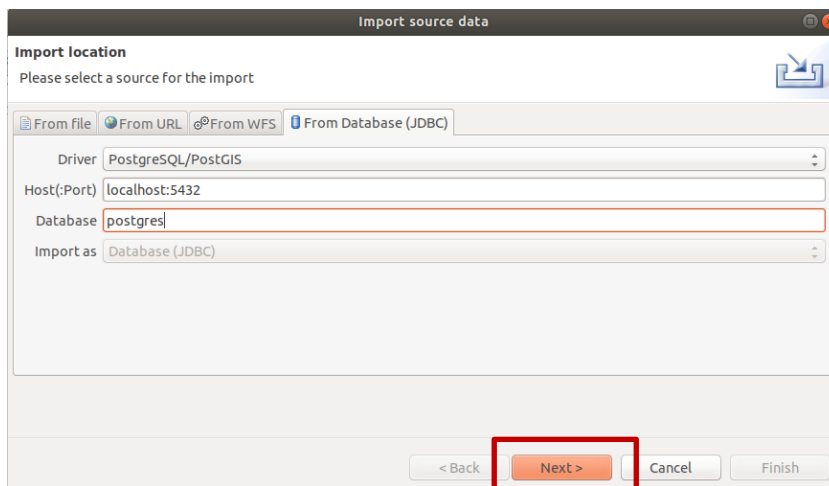
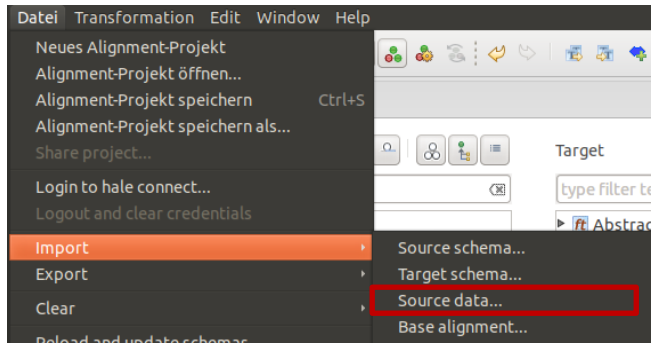
Passwort: postgres

- Laden des Source Schema & Datensatz:
adr_ps_4326 in der Postgisdatenbank postgres im Schema public

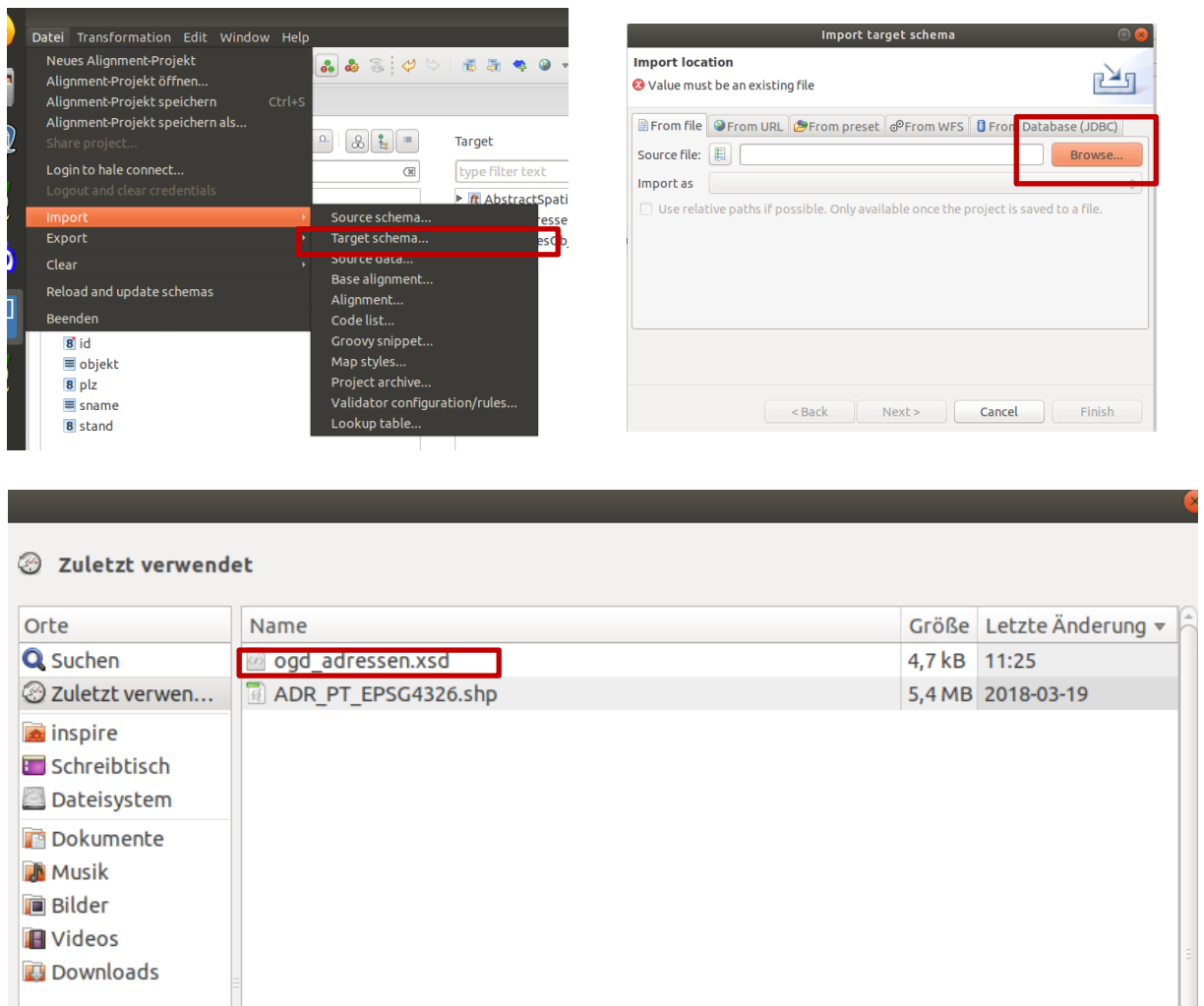




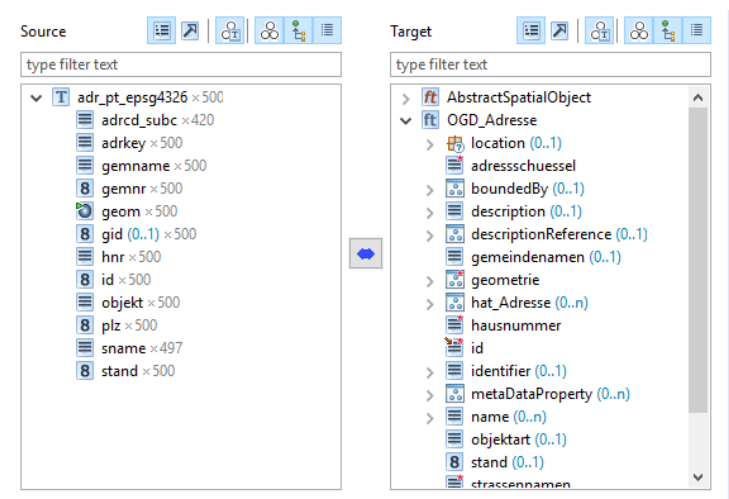
- **Laden des Datensatzes**



- Laden des Ziel-Schema: OGD_Adressen.xsd (zuvor erstellte Datei)

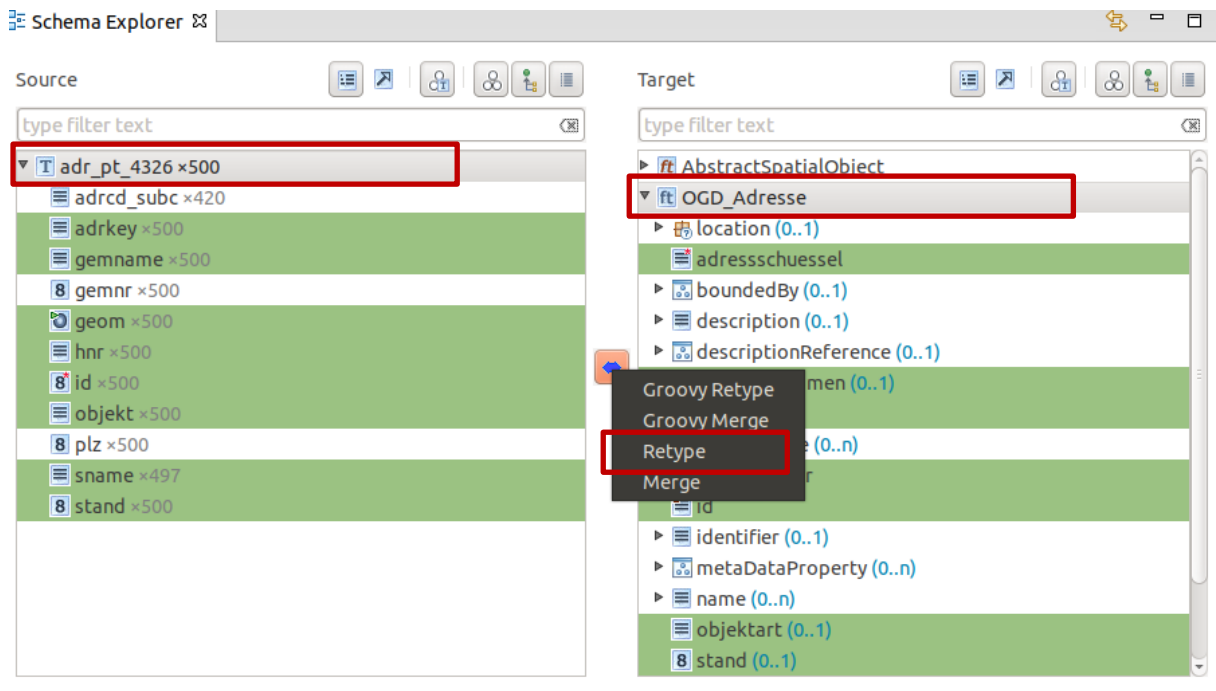


Nach Abschluss sollten die Schemata und die Instanzen wie folgt geladen vorliegen



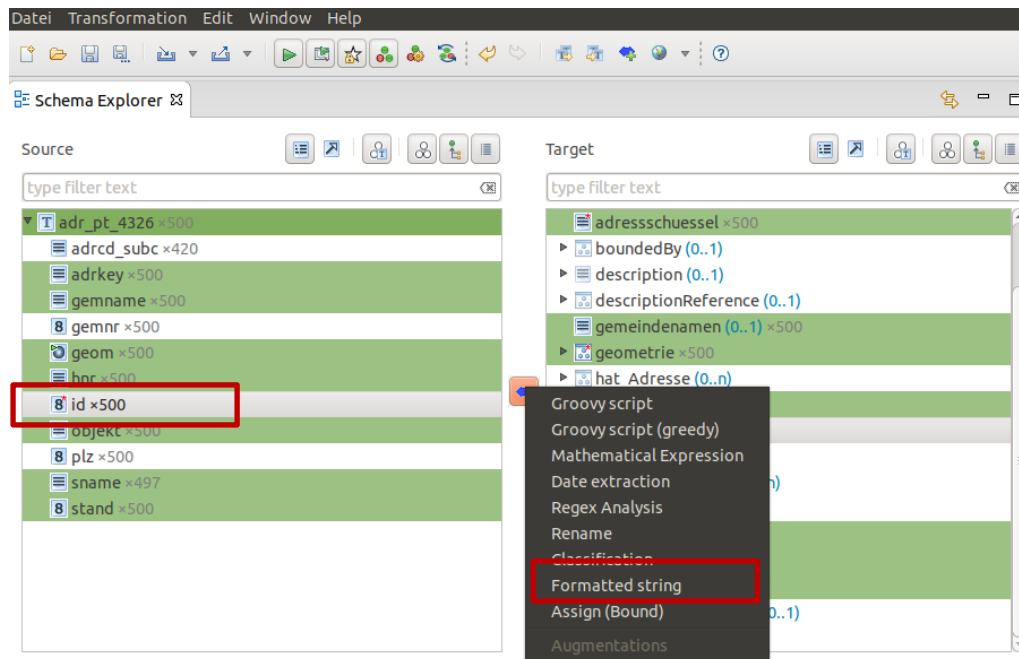
2.2) Mapping der Elemente wie unten angeführt

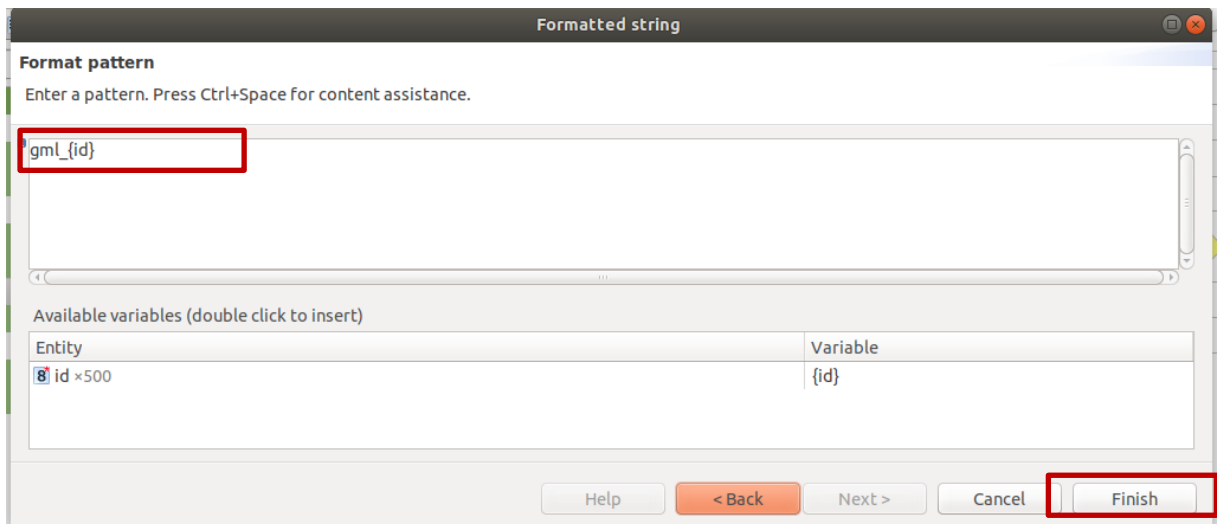
Retype der Tabelle „adr_pt_4326 auf den Feature Type OGD_Adressen



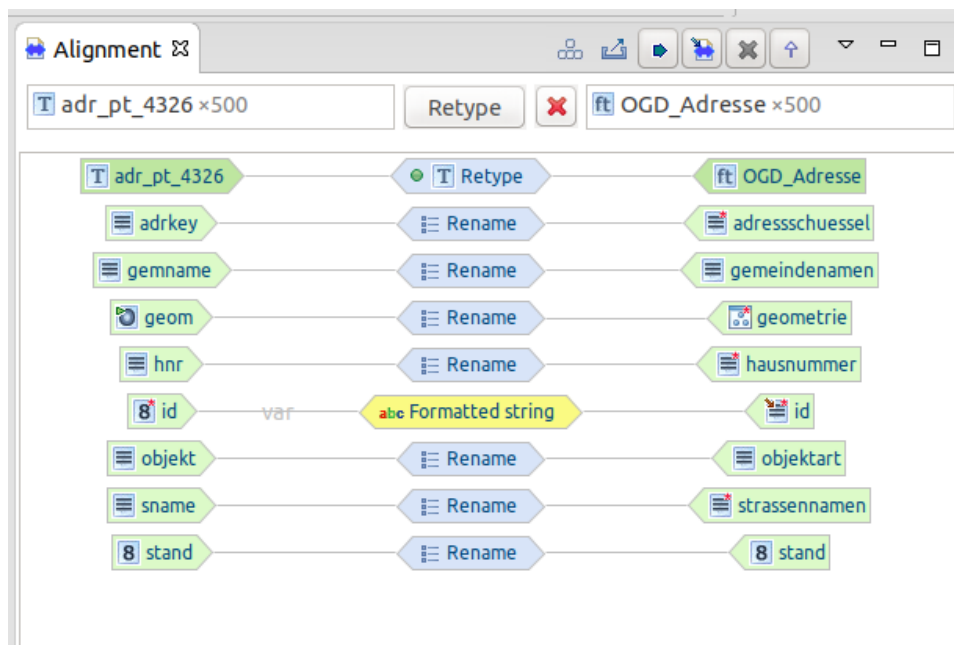
Es wird nur die Rename Funktion und für die GML-ID die Formatted String Funktion benötigt

GML-ID Mapping: select id Tabelle → Auswahl ID Ziel Schema → Transformation mit Funktion “Formatted String

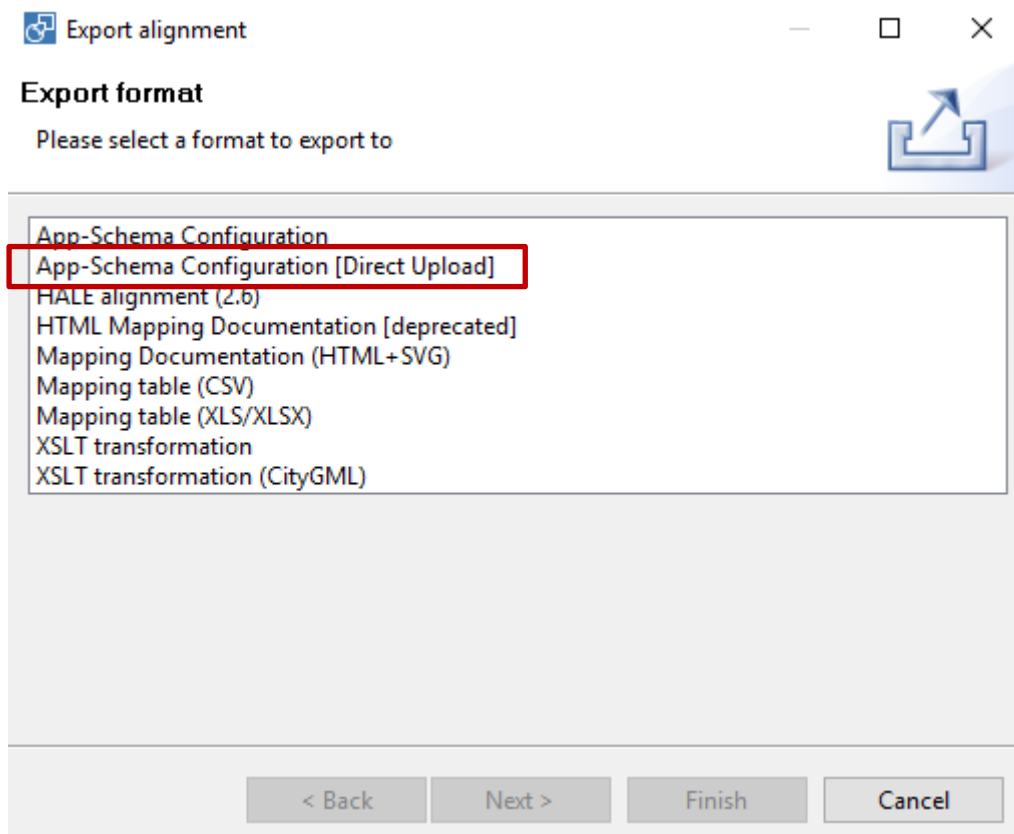
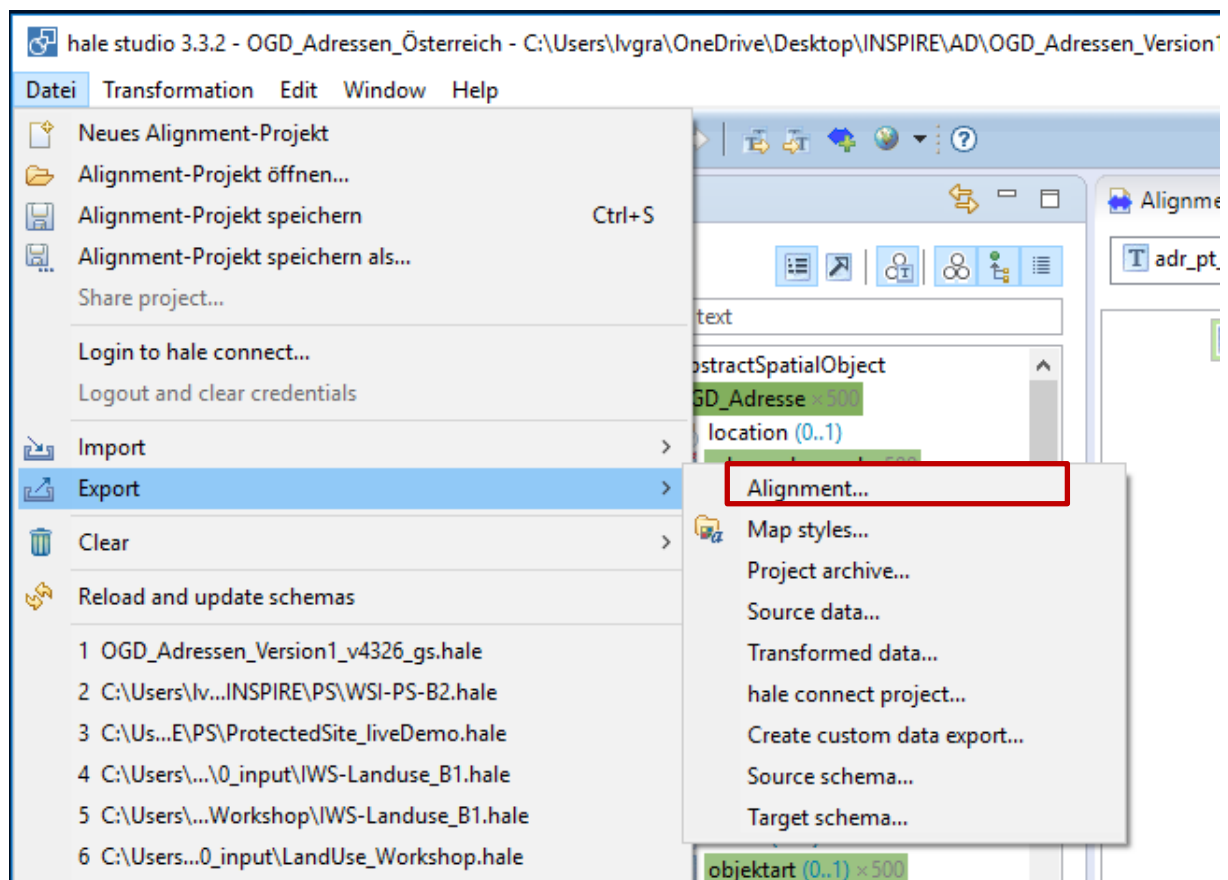




Mappen aller anderen verfügbaren Eigenschaften mit der Funktion Rename



2.4.) Exponieren des harmonisierten Datensatzes mittel Geoserver



Export alignment

Export destination

Target URL

Content type

< Back Next > Cancel Finish

Export alignment

Include target schema

Specify whether the target schema should be included in the exported configuration archive

Include schema

Include target schema in the archive

< Back Next > Cancel Finish

Export alignment

Configure workspaces

If needed, edit the name of a workspace and mark it as isolated to avoid name clashes with feature types already published in GeoServer.

Name	Isolated	Namespace	Features
adr	<input checked="" type="checkbox"/>	www.grillmayer.eu/schema/	OGD_Adresse

< Back Next > Cancel Finish

Export alignment

App-Schema DataStore configuration

Specify PostGIS datastore parameters

Host(:Port) localhost:5432

Database postgres

Schema public

Username postgres

Password postgres

Expose primary keys

< Back Next > Cancel Finish


Export alignment

Authentication

Please enter user name and password to access GeoServer REST services

User: admin

Password: geoserver

 User and password may be saved in the project configuration as plain text. Be aware of this when distributing the project.

< Back Next > Finish Cancel

Das fertige Projekt ist unter: \Desktop\INSPIRE\AD\ogd_adressen_geoserver.halex verfügbar.