Advanced Feature Descriptions User Manual v1.0

Part 1: Karma Modeling

Description: This tutorial steps through creating a R2RML schema to ontology mapping file for the sample USGS Structures DC shapefile.

Karma “Import Ontologies” Instructions:

(NOTE: Easiest way to perform import is to simply copy ontologies into KARMA_HOME/preloaded-ontologies folder; then start/bounce Tomcat)

1. Import the following “core” ontologies:
   (a) GeoSPARQL
   (b) W3C Basic Geo
   (c) Geonames
   (d) DBPedia
   (e) Dublin Core Elements
   (f) Dublin Core Terms
   (g) Dublin Core Types

2. Import the USGS “domain” ontologies:
   (a) USGS TopoNamedPoints

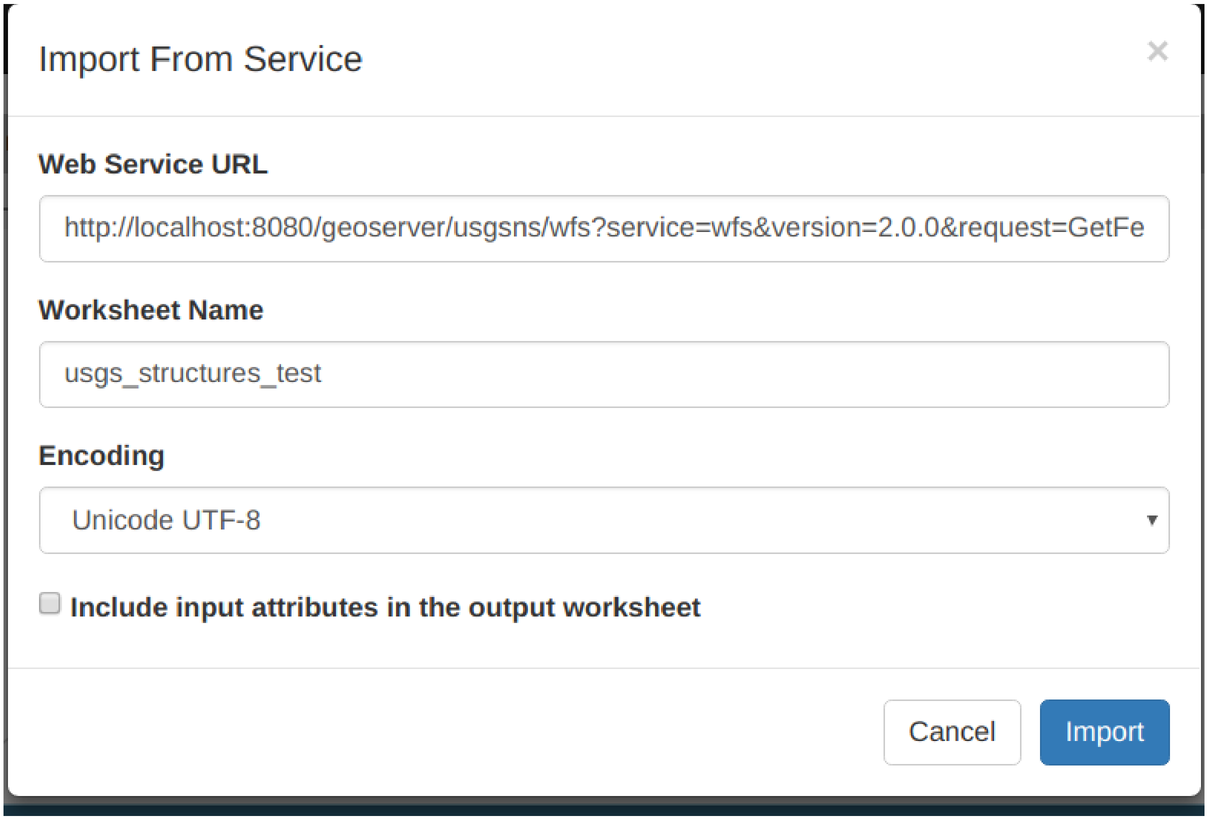
Karma Modeling Instructions:

1. **Import the USGS Structures dataset into Karma from GeoServer:**

   (a) Import → From Service
   (b) Web Service URL:

   http://localhost:8080/geoserver/usgsns/wfs?service=wfs&version=2.0.0&request=GetFeature&featureID=usgs_structures.2

   (c) Worksheet Name: usgs_structures
   (d) Encoding: Unicode UTF-8
   (e) Press “Import” button



Import From Service

Web Service URL

http://localhost:8080/geoserver/usgsns/wfs?service=wfs&version=2.0.0&request=GetFe

Worksheet Name

usgs_structures_test

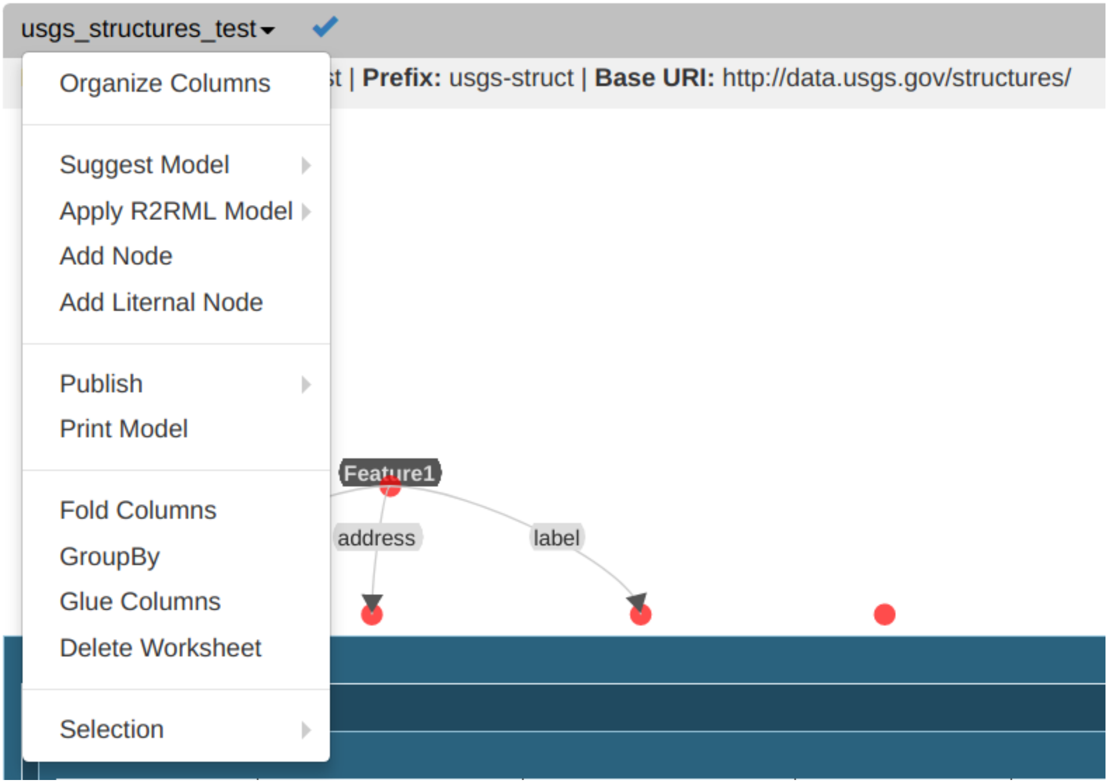
Encoding

Unicode UTF-8

☐ Include input attributes in the output worksheet

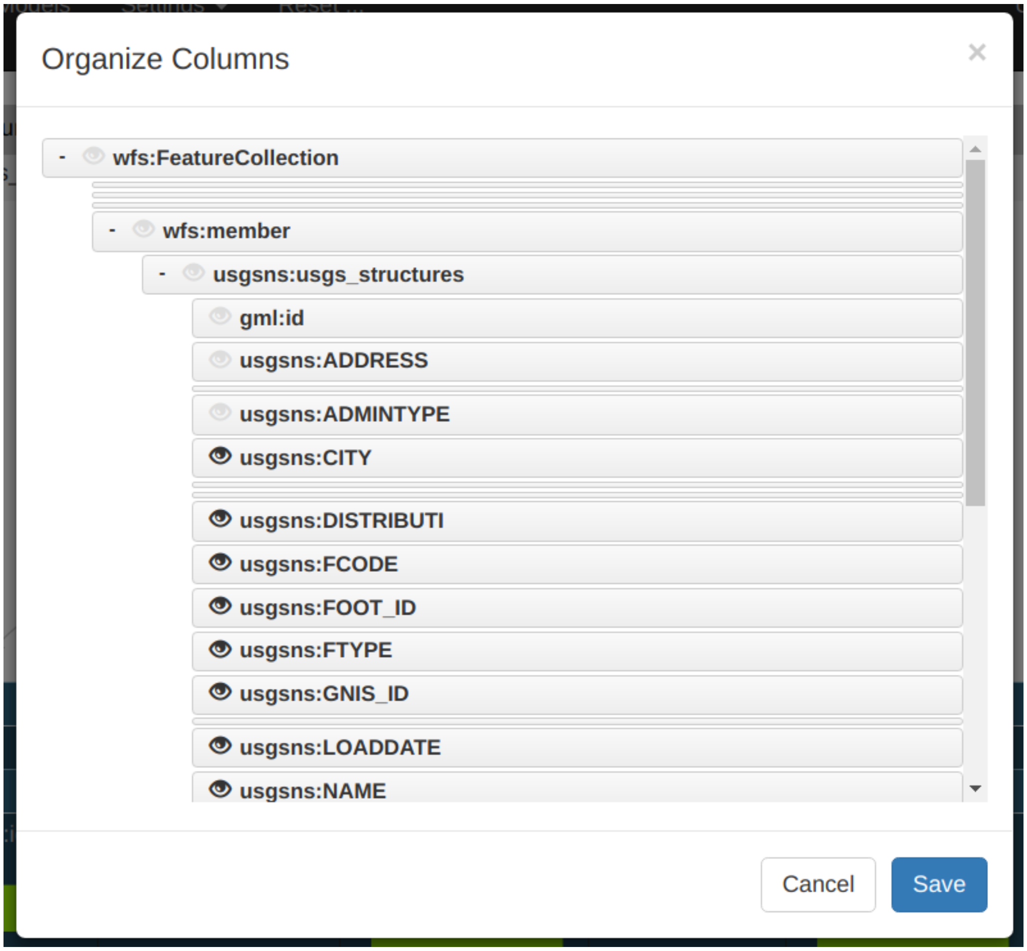
2. **Organize/Hide the columns:**

   (a) Select the drop down menu arrow next to “usgs_structures” at top left of screen after the data has finished importing in Step #1 above.
   (b) Select “Organize Columns”
   (c) Click on the following to hide unneeded columns (i.e. - that won’t get mapped):
      i. numberMatched
      ii. numberReturned
      iii. timeStamp
      iv. ADDRESSBUI
      v. COMPLEX_ID
      vi. DATA_SECUR
      vii. ISLANDMARK
      viii. PERMANENT
      ix. POINTLOCATION
      x. SOURCE_D_1
      xi. SOURCE_FEA
      xii. xmlns:gml
      xiii. xmlns:usgsns
      xiv. xmlns:wfs
      xv. xmlns:xs
      xvi. xmlns:xsi
      xvii. xsi:schemaLocation



usgs_structures_test

…t | **Prefix:** usgs-struct | **Base URI:** http://data.usgs.gov/structures/



Organize Columns

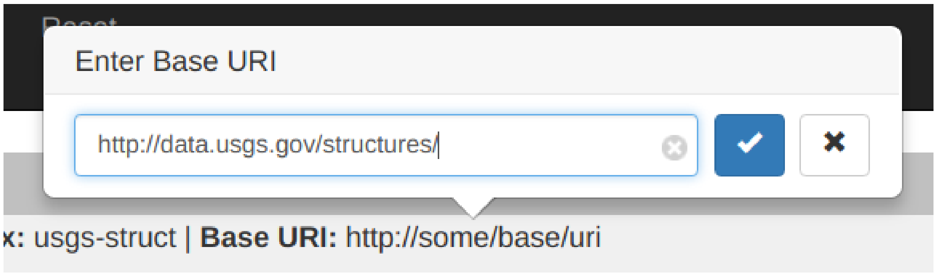
- - wfs:FeatureCollection
  - - wfs:member
    - - usgsns:usgs_structures
      - gml:id
      - usgsns:ADDRESS
      - usgsns:ADMINTYPE
      - usgsns:CITY
      - usgsns:DISTRIBUTI
      - usgsns:FCODE
      - usgsns:FOOT_ID
      - usgsns:FTYPE
      - usgsns:GNIS_ID
      - usgsns:LOADDATE
      - usgsns:NAME

3. Set primary namespace URI and prefix:

(a) To set namespace prefix click on (URI) value in “Base URI”; type desired namespace like:

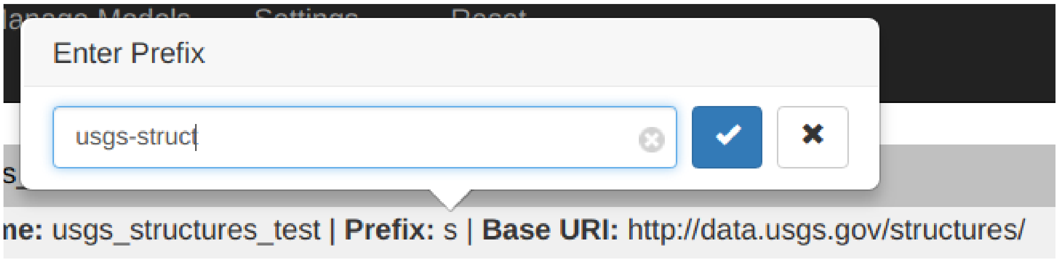
http://data.usgs.gov/structures/

(b) Press Enter or the blue arrow to finish
(c) Next, to set namespace prefix click on value in “Prefix”; type prefix like: usgs-struct
(d) Press Enter or the blue arrow to finish



http://data.usgs.gov/structures/

x: usgs-struct | **Base URI:** http://some/base/uri



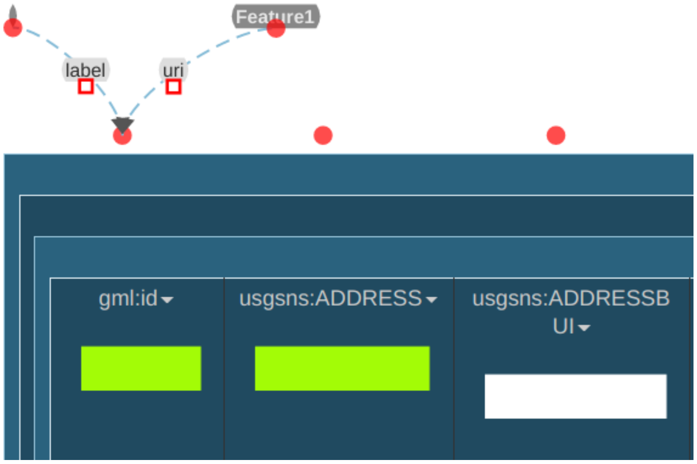
usgs-struct

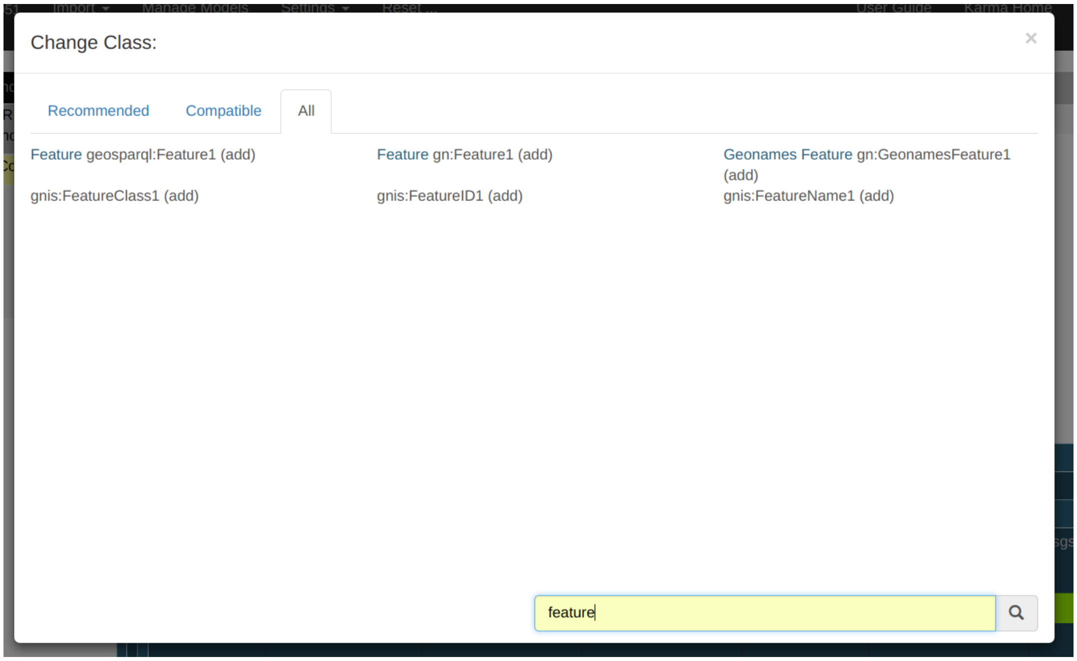
ne: usgs_structures_test | **Prefix:** s | **Base URI:** http://data.usgs.gov/structures/

4. Map gml:id as the URI of a geosparql:Feature:

(a) Hover over red dot above column gml:id and press “Suggest”
(b) Select the gray “leaf” on the red node at other end of graph arc
(c) In the search box at the bottom of the “Change Class” form type “feature” to filter list
(d) Only a few selections should now be available. Select “Feature geosparql:Feature1”
(e) Hover over the arc and select “label”
(f) Under the “All” tab select the first item, “uri of Feature1”; Karma will append the value of this field to the “Base URI” specified in Step #3 above
(g) IMPORTANT: Replace gml:id field with full URI. This = URI used for the feature.
  i. Click on drop down arrow next to “gml:id” column and select PyTransform.
  ii. Modify the code to “return structures_uri(getValue("gml:id"))” and “Save”.
  iii. This appends the value in the field to a namespace (needed for WFS data):

http://data.usgs.gov/structures/usgs_structures.2





Change Class:

Feature geosparql:Feature1 (add)

Feature gn:Feature1 (add)

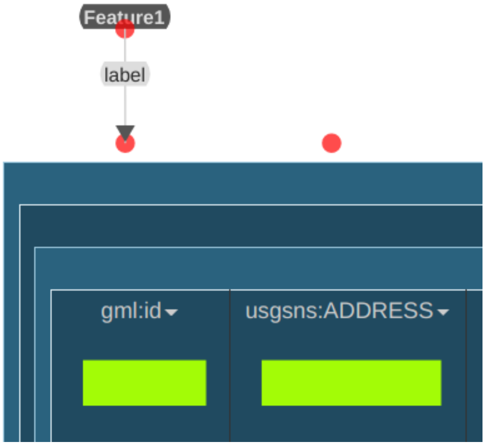
Geonames Feature gn:GeonamesFeature1 (add)

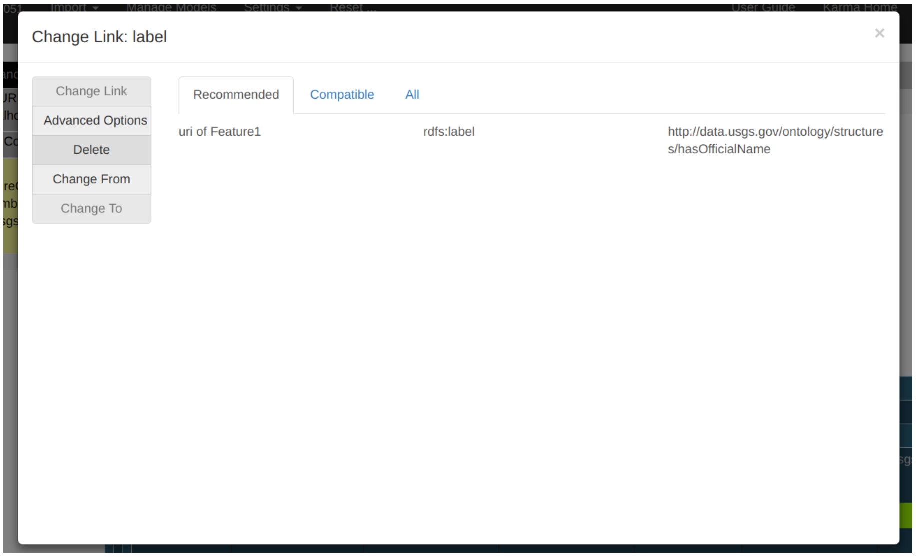
gnis:FeatureClass1 (add)

gnis:FeatureID1 (add)

gnis:FeatureName1 (add)

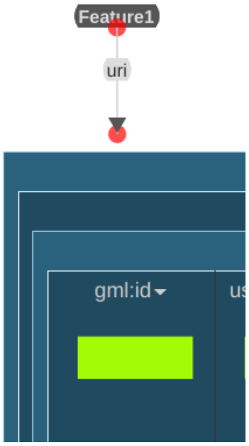
feature





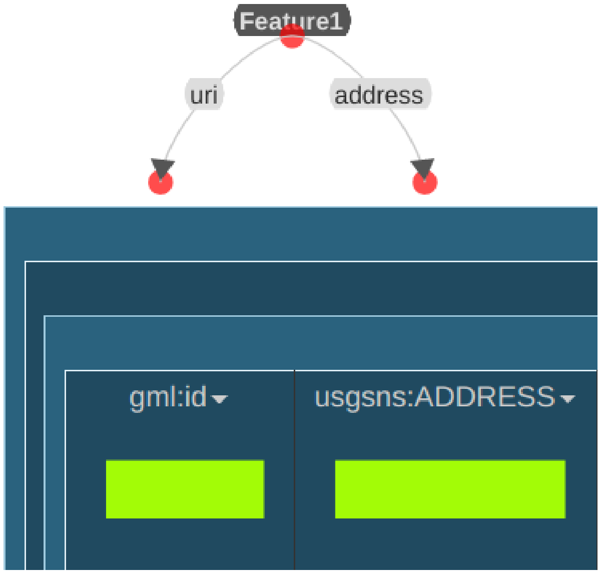
Change Link: label

uri of Feature1 rdfs:label http://data.usgs.gov/ontology/structures/hasOfficialName



5. **Map usgsns:ADDRESS to dbpedia:address property:**

   (a) Hover over red dot above column “usgsns:ADDRESS” and press “Suggest”
   (b) Hover over the arc attached to “Feature1” and select “label”
   (c) In the search box at the bottom of the “All” tab type “address” to filter list
   (d) Select “address http://dbpedia.org/ontology/address”



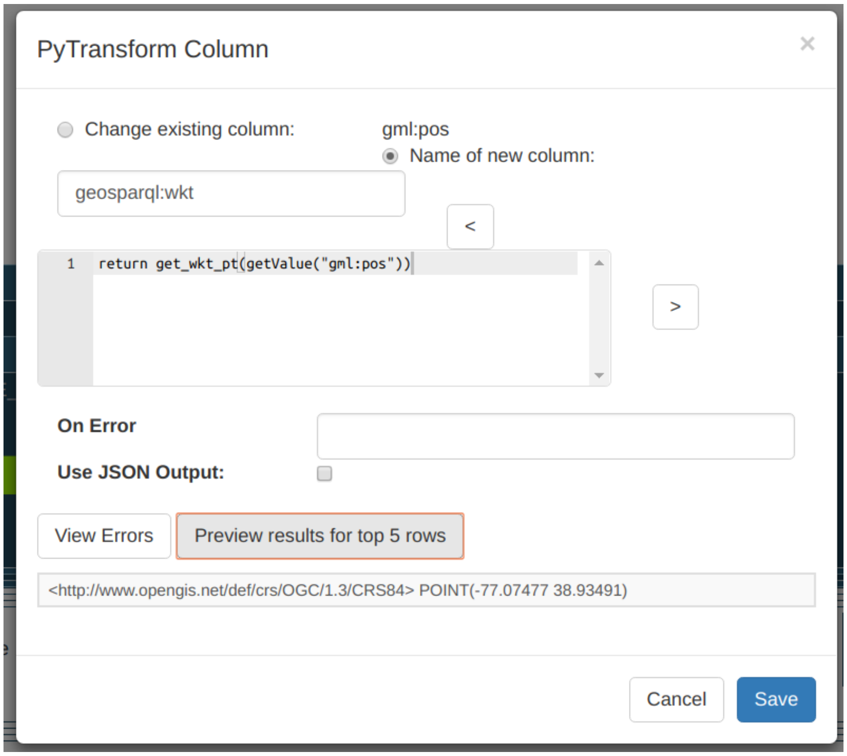
6. **Map usgsns:ADMINTYPE to dbpedia:administrativeDistrict property:**
   (a) Hover over red dot above column “usgsns:ADMINTYPE” and press “Suggest”
   (b) Select the gray “leaf”on the red node; Then select Feature in popup form.
   (c) Click on the drop down arrow directly under the column name “usgsns:ADMINTYPE”
   (d) Select “PyTransform” in the popup form
   (e) Click on “Change Existing Column”
   (f) In the Python textbox paste in the following:

   return get_admin_type(getValue("usgsns:ADMINTYPE"))

   This will execute a Python function, called get_admin_type(), located in KARMA_HOME/python/usgs_structures.py. The function returns a human-readable ‘administrative type’ which is more intuitive for the user (vs. an integer representing an administrative type.
   (g) Click “Preview results for top 5 rows” to see an example calculated. If no results show, click “View Errors” to see if an error occurred.
   (h) If function returned correct results, click ‘Save’
   (i) Next, click on the property name and change it to dbpedia:administrativeDistrict

7. **Map usgsns:ADMINTYPE to mapping from dbpedia:administrativeDistrict:**

(a) Click on “gml:pos” dropdown and select “PyTransform”
(b) Fill in the popup form as shown in image below:



PyTransform Column

○ Change existing column: gml:pos
● Name of new column: geosparql:wkt

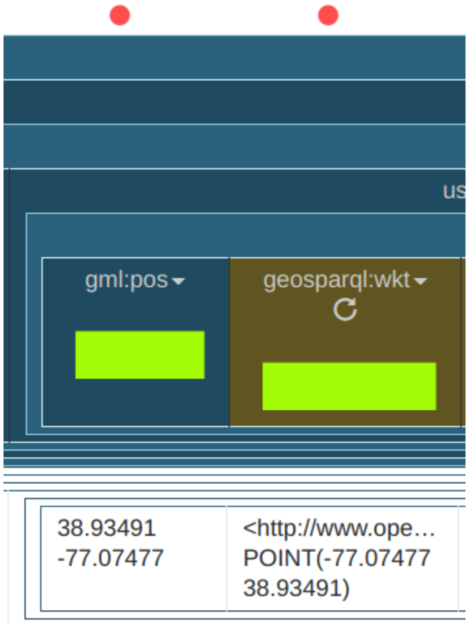
```
return get_wkt_pt(getValue("gml:pos"))
```

On Error

Use JSON Output: □

```
<http://www.opengis.net/def/crs/OGC/1.3/CRS84> POINT(-77.07477 38.93491)
```

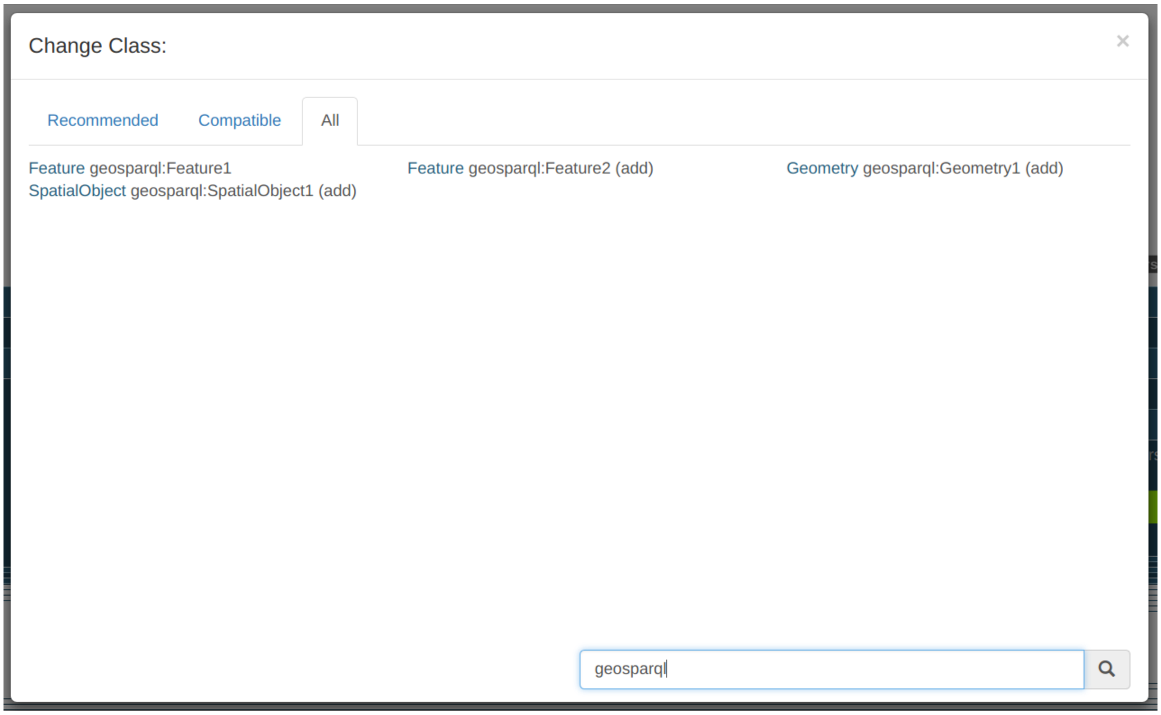
(c) Once ‘Save’ is pressed, the new column should look like the following image:



gml:pos	geosparql:wkt
38.93491 -77.07477	<http://www.ope… POINT(-77.07477 38.93491)

(d) Finally, click on the red dot and press “Suggest”

(e) Select the gray “leaf” on the unlinked red node.
(f) Select geosparql:Geometry1 in the popup as follows:

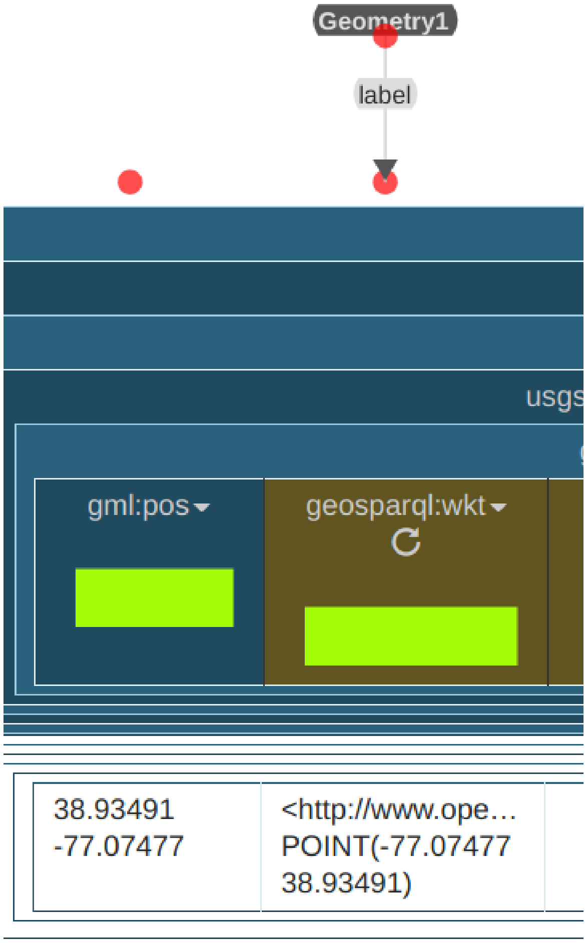


Change Class:

Feature geosparql:Feature1
Feature geosparql:Feature2 (add)
Geometry geosparql:Geometry1 (add)
SpatialObject geosparql:SpatialObject1 (add)

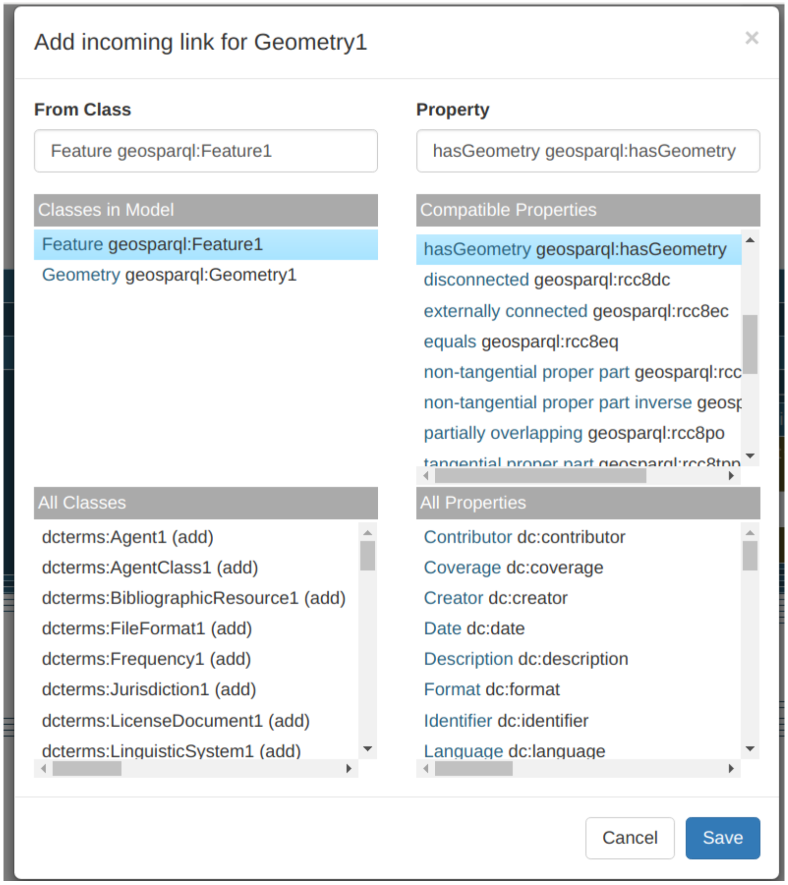
geosparql

(g) The mapping should now look like this:



gml:pos	geosparql:wkt
38.93491 -77.07477	<http://www.ope… POINT(-77.07477 38.93491)

(h) Next, select Geometry1 property. Edit it via selecting geosparql:asWKT as the property.
(i) Finally, add a link from Feature1 to Geometry1. Start by clicking on Geometry1 class.
(j) Click on “Add Incoming Link”. Select Feature1 as the class where link is coming from.
(k) Then set “geosparql:hasGeometry” as the property. See image below for reference:



Add incoming link for Geometry1

From Class

Feature geosparql:Feature1

Property

hasGeometry geosparql:hasGeometry

Classes in Model

- Feature geosparql:Feature1
- Geometry geosparql:Geometry1

Compatible Properties

- hasGeometry geosparql:hasGeometry
- disconnected geosparql:rcc8dc
- externally connected geosparql:rcc8ec
- equals geosparql:rcc8eq
- non-tangential proper part geosparql:rcc
- non-tangential proper part inverse geosp
- partially overlapping geosparql:rcc8po
- tangential proper part geosparql:rcc8tpp

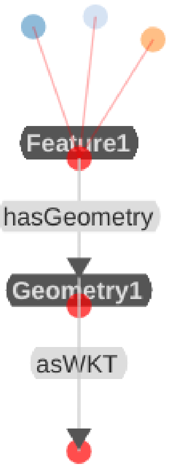
All Classes

- dcterms:Agent1 (add)
- dcterms:AgentClass1 (add)
- dcterms:BibliographicResource1 (add)
- dcterms:FileFormat1 (add)
- dcterms:Frequency1 (add)
- dcterms:Jurisdiction1 (add)
- dcterms:LicenseDocument1 (add)
- dcterms:LinguisticSystem1 (add)

All Properties

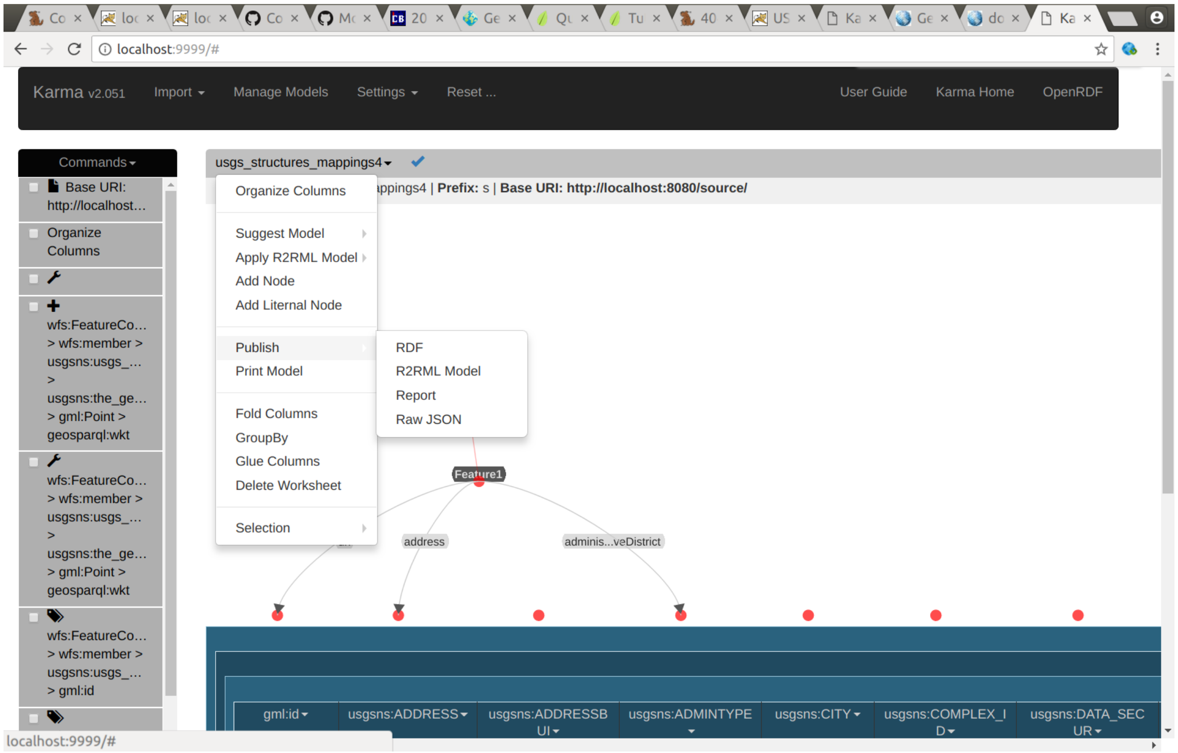
- Contributor dc:contributor
- Coverage dc:coverage
- Creator dc:creator
- Date dc:date
- Description dc:description
- Format dc:format
- Identifier dc:identifier
- Language dc:language

(l) Lastly, the mapping should look like the following:

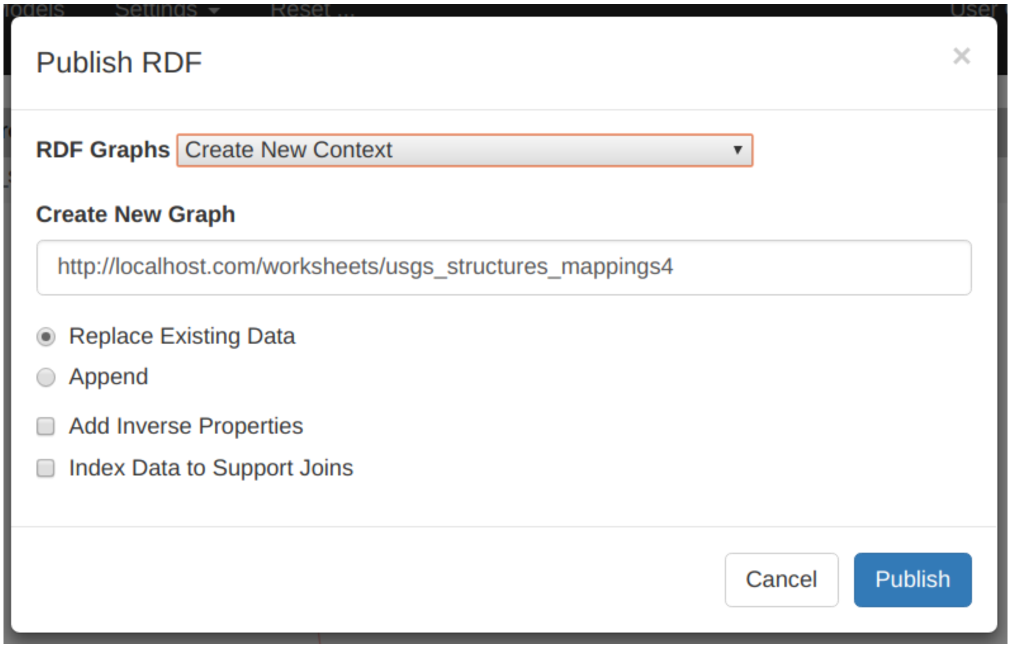


8. Export as RDF file:

(a) To publish the data as RDF, click on the drop down near the top left region where the name of the mapping is listed (“usgs_structures_mapping4” in image below).



(b) Go to: “Publish → RDF”
(c) Fill out popup form as shown below:


Publish RDF

RDF Graphs Create New Context

Create New Graph

http://localhost.com/worksheets/usgs_structures_mappings4

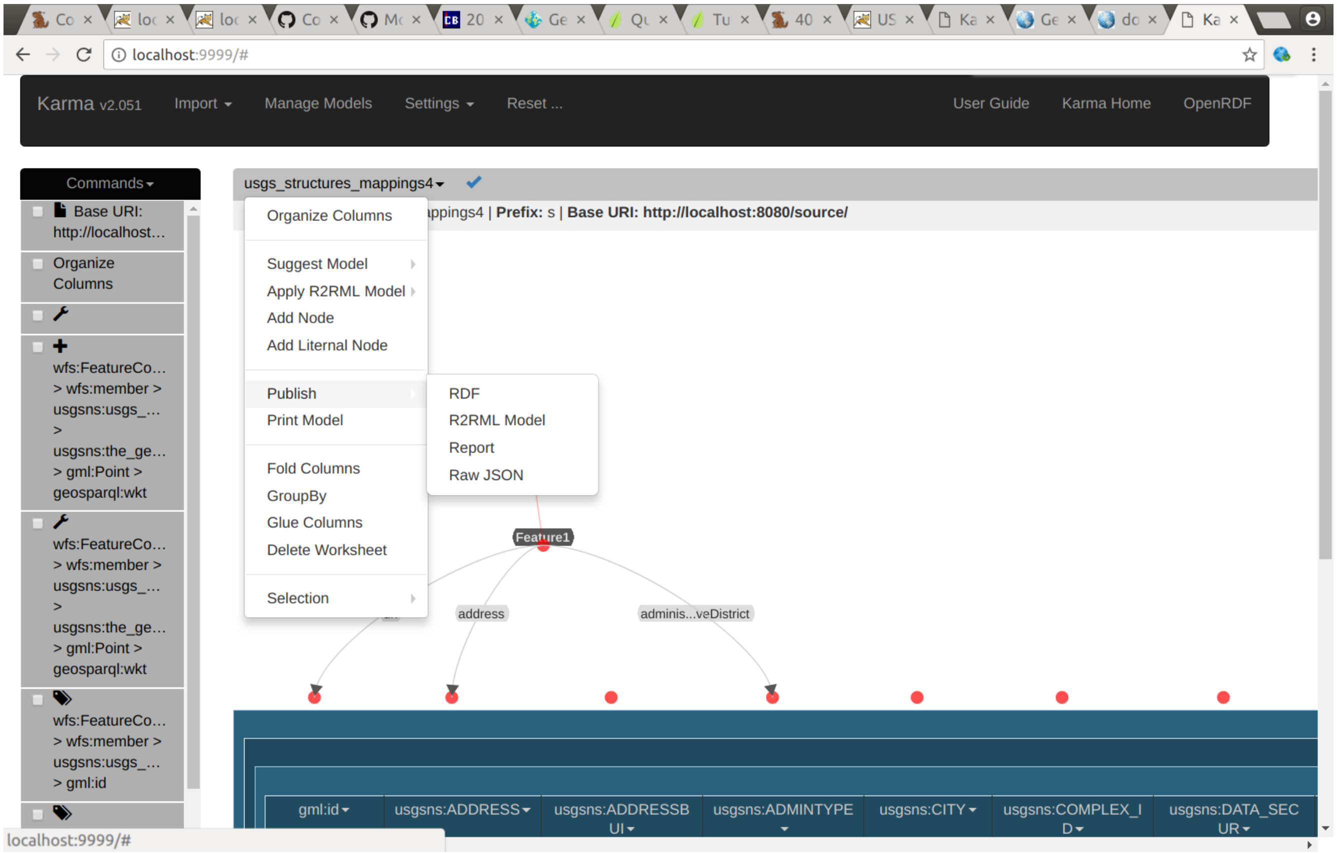
● Replace Existing Data
○ Append

□ Add Inverse Properties
□ Index Data to Support Joins

(d) Click on “RDF to the right of the mapping name drop down you clicked on in step (a)
(e) A Turtle RDF file will open in another tab in browser.

9. Once RDF is correct, you can then export a R2RML mapping file:

(a) To publish the data as RDF, click on the drop down near the top left region where the name of the mapping is listed (“usgs_structures_mapping4” in image below).



(b) Go to: “Publish → R2RML Model”
(c) Click on “RDF to the right of the mapping name drop down you clicked on in step (a)
(d) A R2RML file will open in another tab in browser.

10. Save R2RML file and update Karma-WFS Plugin Configurations:

Following from step 9d, save R2RML mapping file in the following folder:
(This will allow Karma As A Service to translate the Structures GML into RDF)

TOMCAT_HOME/webapps/examples/usgs/r2rml/usgs_structures_r2rml.ttl

11. Update Karma-WFS Plugin configuration files to support new R2RML mappings for USGS Structures data:

(a) The Karma-WFS plugin will, for example, try to dereference a USGS structure geographic feature with the following URI:

http://data.usgs.gov/structures/usgs_structures.123

The “structures” portion of the URI tells the Karma-WFS plugin that the Karma RDF translation service will need the R2RML file saved above in step #10.

(b) Therefore, in the MARMOTTA_HOME/config/karma_wfs_mappings.properties file add the following line:

structures = http://localhost:8080/examples/usgs/r2rml/usgs_structures-model.ttl

(c) The Karma-WFS plugin also will need the base URL of the WFS server hosting the USGS Structures dataset. Thus, append the following line to MARMOTTA_HOME/config/karma_wfs_uris.properties:

structures = http://localhost:8080/geoserver/usgsns/wfs?service=wfs&version=2.0.0&request=GetFeature&featureID=

NOTE: For this demo all GIS data is hosted in the same Geoserver WFS. However, this step allows to aggregate data from multiple distributed WFS servers.

(d) Lastly, the Leaflet Map UI uses a JSON file to lookup namespaces so it can build the Marmotta URL to dereference.

Append this key/value pair:

```
"usgs_structures" : "structures/"
```

To the following file:

```
{tomcat_webapps}/examples/usgs/afd/afd-nsids.json
```

DONE!!!

At this point all configuration for using the USGS Structures dataset is complete and can be used in the Advanced Feature Description System.