Advanced Feature Descriptions

System Manual v1.0

Advanced Feature Description (AFD) System Manual Description:

This document contains the instruction needed to install and run the Advanced Feature Description system which aggregates and fuses attributes of a given geographic feature across multiple, disparate web data sources.

This document does NOT review how to use Karma or LIMES which are the preprocessing tools used to produce schema mapping rules and co-references, respectively. Both Karma and LIMES are described in the USER manual.

This document in split into three parts:

1. System Overview
2. System Installation
3. System Configuration

Background

There are two preprocessing steps that must be performed before the AFD system can aggregate disparate attributes of geographic features:

1. Entity resolution - identifies the same USGS National Map feature in disparate web datasets. Once performed, the resulting co-references are stored in the Marmotta RDF store.

2. Schema alignment - maps the heterogeneous schemas of disparate web data sources into a common view familiar to the user. Once performed, the resulting schema mappings are stored as R2RML Turtle files in the “examples/usgs/r2rml” folder.

More details can be found in the 'Karma' and 'LIMES' sections of the User Manual.

After the co-references and schema mappings have been computed and stored, the user can use the Leaflet-based map UI to display aggregated attributes of National Map features. However, for the prototype, all of the co-references and schema mappings have already been precomputed for the sample data. Therefore, the system is ready to run "out-of-the-box". See the 'Installation' section below.

Downloads

Before proceeding, make sure the needed files have been downloaded from GitHub repo:

https://github.com/tpehle/karma4marmotta

OR the internal USGS repo

1. System Overview

System Software

AFD contains/runs the following ***open source*** software:

1. **VMWare Workstation 12 Player** (12.5.2 build-4638234) - The entire AFD prototype runs within a single VM for convenience. In a production system it would be split over multiple servers/VMs for better performance.
2. **Ubuntu 16.04** – All AFD software runs within a single OS instance.
3. **Java 8 (Oracle JDK 1.8.0_121)** – All AFD software runs on Java (with the exception of Python scripts run by Karma).
4. **Tomcat 8.5.12** – All AFD software runs as webapps within Tomcat.
5. **Geoserver 2.10.2** – Hosts AFD geospatial data and publishes as OGC WFS. It's deployed as a war file within Tomcat.
6. **Karma 2.052** – A user modeling tool used as a **preprocessing tool** (vs. runtime) to map data schemas to ontologies. For AFD the input is a Geoserver WFS map layer. The output is an R2RML mapping file used by Karma As A Service. It's deployed as a war file within Tomcat.
7. **LIMES 1.1.2** – A user modeling **preprocessing tool** used to generate co-references between two datasets. The output is a set of "owl:sameAs" RDF triples which are simply inserted into Marmotta. The service applies the R2RML file to the WFS GML and translates/returns the data as RDF. It's deployed as a war file within Tomcat.
8. **Karma As A Service** – A **runtime tool** that takes as input a Geoserver WFS URL for a geographic feature along with a R2RML mapping file URL. The service applies the R2RML file to the WFS GML and translates/returns the data as RDF. It's deployed as a war file within Tomcat.
9. **Marmotta 3.3.0** – A **runtime triple store** that stores RDF and transparently dereferences RDF from Linked Data sources. For data coming from non-RDF data sources, Marmotta Plugins can be utilized (such as the AFD KarmaWFS plugin). The service applies the R2RML file to the WFS GML and translates/returns the data as RDF. It's deployed as a war file within Tomcat.

NOTE: All of the needed war files are located in:

afd/wars

(a) *geoserver.war (Geoserver)*
(b) karma-web-2.052.war (Karma)
(c) marmotta.war (Marmotta)
(d) *web-services-rdf-0.0.1-SNAPSHOT.war (Karma As A Service)*

AFD contains the following ***custom*** software:

1. **Leaflet Map** (UI) – The map source code is all HTML and Javascript. It is located in:

   afd/src/ui/

2. **Karma WFS Marmotta Plugin** (WFS GML-to-RDF translation) – This plugin is written as a Java library. It is located here:

    afd/src/karma-wfs/

System Data

The AFD prototype contains the following ***sample geospatial data (shape files)***:

1. **USGS GNIS – DC**: Placenames in Washington D.C.

    afd/data/shp/usgs_layers/gnis/DC_Features_20180401.shp

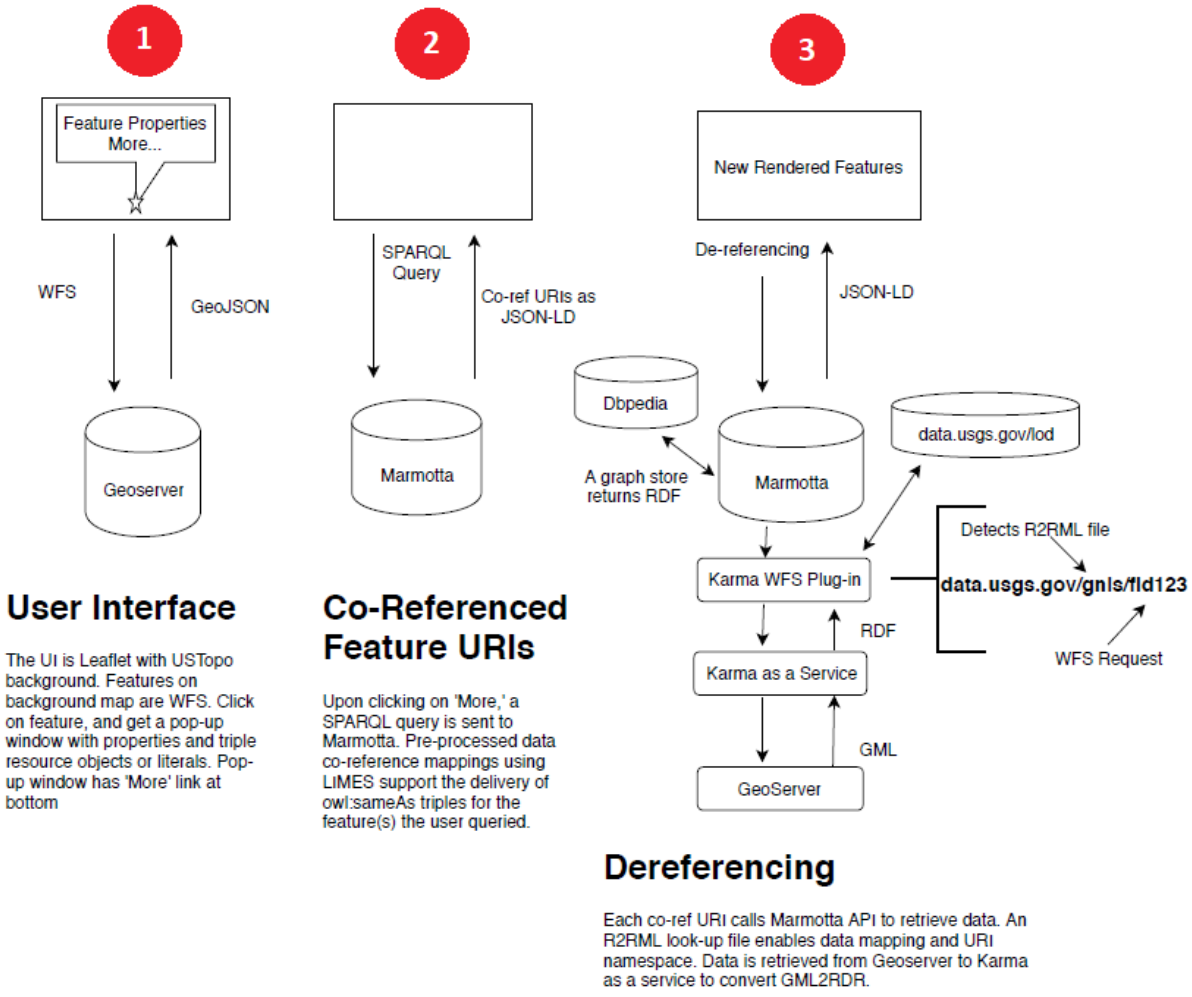
2. **USGS NTD Structures – DC**: Structures in Washington D.C.

    afd/data/shp/usgs_layers/structures/Struct_Point.shp

3. **Geonames.org – DC**: Placenames in Washington D.C.

    afd/data/shp/geonames/geonamesDC.shp

System Architecture



The image above describes the **three primary operations** that the **AFD runtime** performs:

1. The Leaflet Map UI begins by simply performing a typical OGC WFS request for a map layer and displays the map on the UI. A user can click on a feature and attributes are displayed.
2. If the user clicks on an “Advanced Feature Description” link at the bottom of a given attributes popup, the UI performs a simple SPARQL query to Marmotta requesting all of the known co-references for the selected WFS feature URL (defined by the WFS GetFeature operation). Marmotta simply returns URIs of known co-references to the UI.
3. Upon receiving the co-references URIs from Marmotta, the UI asynchronously dereferences each URI via Marmotta.
   a) If the URI starts with “http://data.usgs.gov/”, Marmotta intercepts the URI and uses the KarmaWFS Plugin to obtain the RDF data.
   b) In this case, the URI is parsed to obtain which Geoserver WFS URL to dereference.
   c) Similarly, from the URI it detects which R2RML mapping file URL to use.

d) This information is then used to call Karma As A Service to translate the GML from Geoserver to RDF.
e) Once the RDF is obtained from the KarmaWFS plugin, Marmotta returns the RDF as JSON-LD to the UI.
f) The UI parses the JSON-LD and populates the corresponding attributes and values in the tabs on the right side of the UI.

2. System Installation

Installation Prerequisites

1. Java is installed (JAVA_HOME path set in ~/.bashrc)
2. Tomcat is downloaded and is set to run on default port of 8080 and accessible via http://localhost:8080.

Installation Instructions

1. **Copy ALL .war files into Tomcat "webapps" folder**:

   (a) geoserver.war (Geoserver)
   (b) karma-web-2.052.war (Karma)
   (c) marmotta.war (Marmotta)
   (d) web-services-rdf-0.0.1-SNAPSHOT.war (Karma As A Service)

   from: afd/wars/

   to: TOMCAT_HOME/apache-tomcat-8.5.12/webapps/

2. **Copy the KarmaWFS Plugin jar (ldclient-provider-karma-wfs-3.3.0.jar) into Marmotta**:

   from: afd/lib/

   to: TOMCAT_HOME/apache-tomcat-8.5.12/marmotta/WEB-INF/lib/

3. **Copy the Leaflet-based UI application folder "usgs" into Tomcat "examples":**

   from: afd/src/ui/usgs/

   to: TOMCAT_HOME/apache-tomcat-8.5.12/examples/

   Correct path to usgs.html (Leaflet Map) = {tomcat_webapps}/examples/usgs/afd.html

4. **LIMES files:**

(a) LIMES has been compiled and is located in:

    afd/limes/limes-gui

(b) It can be started via the following command:

afd/limes/limes-gui$ java -jar limes-gui-1.5.5-jfx.jar

(c) LIMES can be copied to another location, but the “lib” folder must be present in the same directory.

(d) A compiled version of LIMES has been delivered for convenience. However, it can also be compiled using Maven. See LIMES website for more info:

https://github.com/dice-group/LIMES

3. System Configuration

AFD System Configuration consists of the following primary steps (detailed in this section):

(a) Configure Geoserver
(b) Configure Marmotta
(c) Configure Web Karma
(d) Configure Leaflet Map UI

Configure Geoserver

1. **Copy the sample geospatial shapefiles into Geoserver data directory:**

(a) Copy ALL files in afd/data/shp/usgs_layers/gnis to:

{tomcat_webapps}/geoserver/data/data/afd

(b) Copy ALL files in afd/data/shp/usgs_layers/structures to:

{tomcat_webapps}/geoserver/data/data/afd

(c) Copy ALL files in afd/data/shp/geonames to:

{tomcat_webapps}/geoserver/data/data/afd

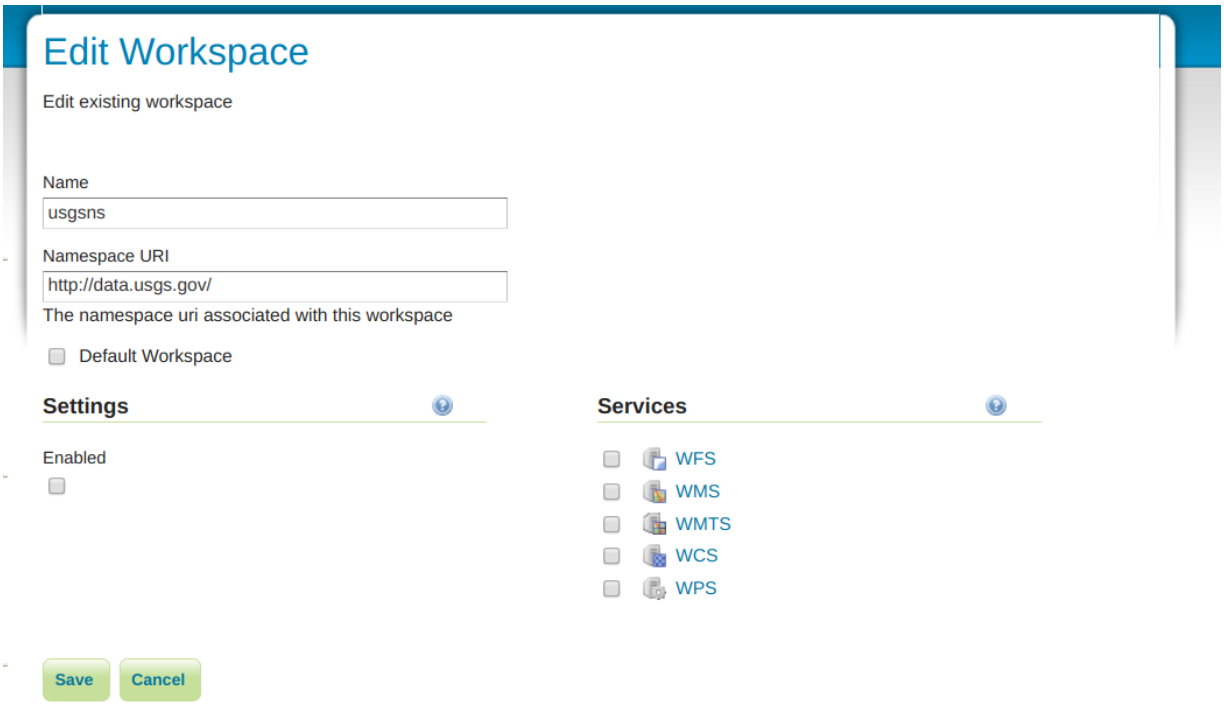
2. **Make sure Tomcat is started. It may take a few minutes to start Geoserver for the first time as it deploys in Tomcat.**

3. **Go to http://localhost:8080/geoserver. Geoserver UI should load. The default login is:**

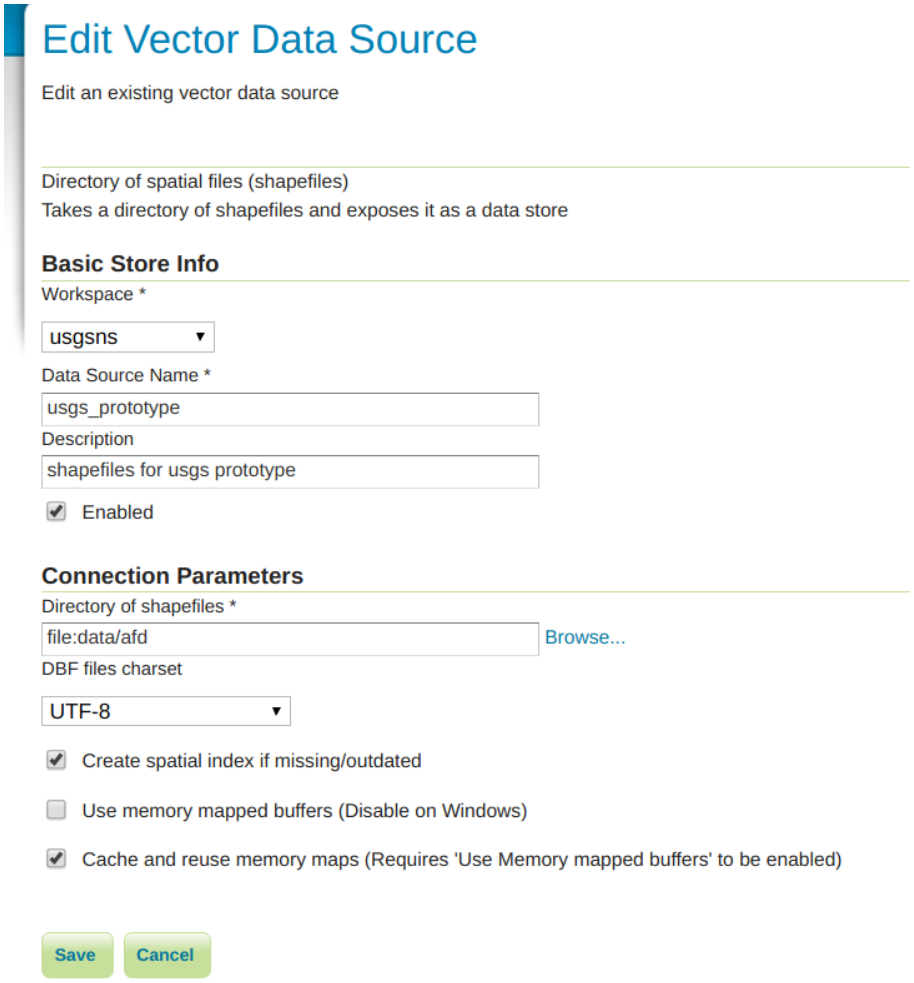
user: admin
password: geoserver

4. **Use the instructions below to load and configure EACH of the map layers (gnis, structures, geonames):**

(a) To load a layer a namespace and datastore must first exist. In Geoserver UI go to: Data → Workspaces in the menu on the left.
(b) If the “usgsns” is not listed, click on “Add New Workspace”. Fill in form and save as shown below.



(c) Next, go to: Data → Stores in the menu on the left.
(d) If “usgs_prototype” is not listed, click on “Add New Store”. Fill in form as shown below. Under “Connection Parameters” browse to and select the folder “data/afd”. Press Save.



(e) Finally, to add a new layer, go to: Data → Layers.
(f) Click on “Add a New Layer”.
(g) From the dropdown choose “usgsns:usgs_prototype”.
(h) Look for the desired shape file name (GNIS_DC_Features_20180401, for example).
(i) If it doesn’t have a check box next to it, it is NOT yet published in Geoserver.
(j) To publish the layer, click on “Publish” link to the right of unpublished layer.
(k) In the new layer form, make sure the Name and Title text boxes match the layer name exactly (for instance, DC_Features_20180401).
(l) Under “Bounding Boxes” section of form, click on “compute from data” to compute the native bounding box.
(m) Also under “Bounding Boxes” section of form, click on “compute from native bounds” to compute the Lat/Lon bounding box.
(n) Scroll to the bottom of the page and press “Save”.

IMPORTANT: The Leaflet map UI is expecting the *name* of the map layers in Geoserver to be *exactly*:

- GNIS_DC_Features_20180401
- usgs_structures
- geonamesDC

5. **To confirm the layers will load in Geoserver, try "Previewing Layer" for each of the above layers using the Geoserver UI.**

6. **Lastly, to confirm the Geoserver URL that AFD will be using works, enter the following URL into a web browser (Firefox, Chrome, etc.):**

http://localhost:8080/geoserver/usgsns/wfs?service=wfs&version=2.0.0&request=GetFeature&featureID=GNIS_DC_Features_20180401.45

WFS GML data should be returned to the browser if working properly.

Configure Marmotta

1. **Set MARMOTTA_HOME in .bashrc in home directory of Linux user launching Tomcat**

MODIFY and append this line to .bashrc profile:

export MARMOTTA_HOME=/home/usgs/afd/software/marmotta_home

Make sure to modify above path to match path to your MARMOTTA_HOME

2. **Bump up Tomcat RAM for Marmotta (and the other web apps in Tomcat):**

export JAVA_OPTS="-Djava.net.preferIPv4Stack=true -Xmx2048m -XX:PermSize=128m
-XX:MaxPermSize=256m -XX:+UseConcMarkSweepGC -XX:+CMSClassUnloadingEnabled
-XX:+UseConcMarkSweepGC -XX:+HeapDumpOnOutOfMemoryError"

NOTE: The -Xmx2048m memory size can be adjusted as needed.

3. **Copy KarmaWFS Marmotta Plugin config files into MARMOTTA_HOME:**

Copy the folder afd/config/marmotta/config to MARMOTTA_HOME/

4. **Edit the "karma_wfs_config.properties" config file IF NEEDED:**

Confirm http://localhost:8080 is used in the line:

karma.url = http://localhost:8080/web-services-rdf-0.0.1-SNAPSHOT/rdf/r2rml/rdf

This URL indicates where Karma As A Service (not “Web Karma”) is located.

5. **Edit the “karma_wfs_mappings.properties” config file IF NEEDED:**

   Confirm http://localhost:8080 is used in the following lines:

   gnis = http://localhost:8080/examples/usgs/r2rml/gnisDC-model.ttl
   structures = http://localhost:8080/examples/usgs/r2rml/structuresDC-model.ttl
   geonames = http://localhost:8080/examples/usgs/r2rml/geonamesDC-model.ttl

   NOTE: These URLs indicate where the R2RML mapping files built in “Web Karma” are hosted.

   IMPORTANT: When a new R2RML mapping file is created in “Web Karma”, the R2RML file should be placed in “{tomcat_webapps}/examples/usgs/r2rml” folder AND an entry should be created in this “karma_wfs_mappings.properties” file. This is required by the KarmaWFS Marmotta Plugin. In addition, add an entry to “{tomcat_webapps}/examples/usgs/afd/afd-nsids.json”.

6. **Edit the “karma_wfs_uris.properties” config file IF NEEDED:**

   Confirm http://localhost:8080 is use in the following lines:

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

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

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

   NOTE: These URLs indicate to KarmaWFS plugin how to construct the Geoserver “GetFeature” service call to retrieve GML for the desired geographic entity.

   IMPORTANT: When a new R2RML mapping file is created in “Web Karma”, this properties file should also be updated to specify the Geoserver URL to KarmaWFS plugin. For this prototype all WFS data is located on the same local server. However, external WFS service URLs can also be added.

7. **Make sure Tomcat is started. It may take a minute to deploy Marmotta the first time. Browse to http://localhost:8080/marmotta. Marmotta UI should load.**

8. **Configure a LD Cache Endpoint in Marmotta for the KarmaWFS Plugin:**

(a) On left menu, go to: Others → Linked Data Caching.
(b) Click on “Configuration” and scroll down to LD-Cache Endpoints.
(c) Under “Add LD-Cache Endpoint” fill out the information below to create a KarmaWFS Endpoint.
   i. Name = Karma Endpoint - USGS WFS
   ii. Kind = karma rdf service
   iii. Prefix = http://data.usgs.gov/*
   iv. Endpoint = http://localhost:8080/web-services-rdf-0.0.1-SNAPSHOT/rdf/r2rml/rdf
   v. Expiry = 240 (# of minutes to store in cache before expiring)

(d) Click on “Upload”. If a login form pops up, enter the Marmotta DB login.

9. In a web browser (Firefox, Chrome, etc.), test dereferencing Karma-mapped RDF:

Goto:

http://localhost:8080/marmotta/meta/application/rdf+xml?uri=http://data.usgs.gov/GNIS_DC_Features_20180401.45

If *Marmotta* + the *KarmaWFS LD Cache Endpoint* + *Karma As A Service* + *Geoserver* are configured correctly, RDF-XML data should be returned/downloaded to the browser.

If this does *not* return data, revisit the steps 1-8 in this subsection (“Configure Marmotta”). Also, search for “http://localhost" in the following files:

(a) {tomcat_webapps}/examples/usgs/r2rml/gnisDC-model.ttl
(b) {tomcat_webapps}/examples/usgs/r2rml/structuresDC-model.ttl

Confirm any of these localhost references have the correct port, 8080.

10. Load precomputed co-references RDF into Marmotta:

(a) Find the two coreferences files in “afd/limes/corefs”.
(b) Follow Step #6 in the LIMES AFD User Manual to insert the co-references into Marmotta.

NOTE: These co-references (and any additional co-references computed by LIMES) must be loaded into Marmotta for AFD aggregation to work in the Leaflet Map UI.

Configure Web Karma

1. **Confirm Karma home environment variable:**

Karma stores all user settings in folder {user.home}/karma by default.

The AFD prototype uses this default location.

From the Karma online documentation:

"You can change this default location by setting the KARMA_USER_HOME environment variable or by clicking on it in the footer of Karma UI and then modifying it."

2. **Import needed ontologies into Web Karma:**

(a) Individual ontologies can be loaded one by one into Karma via the Web Karma UI. However, if a lot of ontologies are needed, this can be cumbersome.
(b) An easier way is simply to copy ALL of the ontologies:

from: afd/karma/preloaded-ontologies

to: KARMA_USER_HOME/karma/preloaded-ontologies

NOTE: On my machine (with user 'usgs') it was here:

/home/usgs/karma/preloaded-ontologies

NOTE: Any additional ontologies needed for mapping can be pasted into this folder.

(c) Bounce Tomcat for ontologies to be imported.

3. **Copy the needed Python scripts into Web Karma:**

Copy each of the Python scripts:

from: afd/karma/python/

to: KARMA_USER_HOME/karma/python

4. **Web Karma configuration is now done. For clarification, Karma As A Service, is simply a .war file deployment. There is NO additional configuration required.**

Configure Leaflet Map UI

1. **Confirm Tomcat is started.**

2. **Confirm the Leaflet Map UI can load:**

(a) In a web browser go to: http://localhost:8080/examples/usgs/afd.html
(b) An Open Street Map layer should load as the basemap. If it doesn't, confirm access to external internet.

3. **Confirm a Geoserver Layer can successfully load.**

(a) In the top menu, go to: Layer → USGS GNIS.

(b) After a few seconds, pushpins over Washington D.C. should display. If not, see Step #4 under the “Configure Geoserver” section above (map layer naming issue in Geoserver).
(c) Click on a pushpin. Attributes of the feature should display in a popup.

4. **Click on “Advanced Feature Description” to aggregate additional information about the feature.**

(a) If Geoserver, Marmotta and Karma configuration steps have been completed above, tabs should appear on the right side of the Leaflet Map UI containing attributes and values for the feature from different data sources.
(b) If no tabs load, try a few different features. Not all features will have co-references available (so no tabs). However, most of the features should at least have a co-reference. So if no tabs are loading for MANY of the map features, then there’s an issue.
(c) To debug, confirm Step #9 in “Configure Marmotta” returns RDF. If it does return RDF, then it’s most likely an issue in UI.
(d) To debug UI issues, there are some references in {tomcat_webapps}/examples/usgs/afd/afd-wfs.js that could cause issues:

// default geometry column name, base wfs url, feature type namespace & crs
var GEOM_FLD = 'the_geom';
var WFS_BASE_URL = 'http://localhost:8080/geoserver/wfs';
var TYPE_NS = 'usgsns';

NOTE: The sample shape files in the prototype all use “the_geom” as the geometry field name, so shouldn’t be an issue; This can be checked, however, in Geoserver Layer page.

NOTE: Confirm http://localhost:8080 is used in the WFS_BASE_URL.

NOTE: Confirm in Geoserver that the namespace for sample data = ‘usgsns’ (all lower case).

(e) There’s also a reference in {tomcat_webapps}/examples/usgs/afd/afd-rdf.js that could also cause issue:

var MARMOTTA_BASE_URL = 'http://localhost:8080/marmotta';

Confirm that http://localhost:8080 is used within this URL.

(f) Lastly, there’s references in {tomcat_webapps}/examples/usgs/afd/afd-nsids.json that could also cause issue:

{

"GNIS_DC_Features_20180401" : "gnis/",
"usgs_structures" : "structures/",
“geonamesDC” : “geonames/”

}

If the three lines above aren’t present in the JSON file, this will cause an issue in the Leaflet Map UI. The UI uses these WFS layer names (like ‘GNIS_DC_Features_20180401’) to help construct the Marmotta URL that will be queried for coreferences like the following example (see bold portion):

*http://data.usgs.gov/**gnis**/{feature_id}*

IMPORTANT: When adding a new dataset into Geoserver (+ its corresponding R2RML mapping file), make sure to append an entry for it in the afd-nsids.json file. The value (for example, “gnis/”) should match the name used in Steps #5 and #6 in “Configure Marmotta” section above.

DONE!!!