OBEOS

Linked Data Prototype
Software Configuration File

Title	:	OBEOS Linked Data Prototype Software Configuration File
Abstract	:	This document describes the procedure to prepare, deploy and configure the constituents of the OBEOS Linked Data Prototype (LDP) software. In particular, this includes the installation of CKAN, the OBEOS LDP extension, an Open-Source Query Resolver service, an RDF Database, and their dependencies.

Table of Contents

List of Figures

List of Tables

1 Introduction

1.1 Purpose of the Document

The version of the document contains instructions for installing and configuring the Linked Data Prototype software.

1.2 References

Table 1 – References

ID	Reference
[EO-LD]	**EO Metadata Discovery using Linked Data** OBEOS-SPB-D3100.1 OGC Discussion Paper referenced OGC 16-074, Version 0.2.0, 25-Mar-2016
[LDP-SUM]	**OBEOS Linked Data Prototype Software User Manual** OBEOS-SA-D5220.9, Issue 1.2.0, 12-Apr-2016
[RD.CKAN]	**CKAN – The Open Source Data Portal Software** http://ckan.org/
[RD.PostgreSQL]	**PostgreSQL Object Relational Database Management System** http://postgresql.org/
[RD.ApacheSolr]	**Apache Solr open-source enterprise search platform** http://lucene.apache.org/solr/
[RD.ApacheServer]	**Apache HTTP Server** https://httpd.apache.org/
[RD.Nginx]	**Nginx HTTP Server** http://nginx.org
[RD.Supervisor]	**Supervisor Process Control System** http://supervisord.org/
[RD.RESTo]	**RESTful EO Product Search Engine (CNES)** https://github.com/jjrom/resto
[RD.iTag]	**iTag footprint tagger (CNES)** https://github.com/jjrom/itag/
[RD.DockerCentOS]	**Procedure for installing Docker on Linux CentOS 7.x** https://docs.docker.com/engine/installation/centos/
[RD.GeoTriples]	**GeoTriples: a Tool for Publishing Geospatial Data as RDF Graphs Using R2RML Mappings** K: Kyzirakos, S: Manegold (CWI), I: Vlachopoulos, D: Savva, and M: Koubarakis (NKUA). In TerraCognita 2014. http://event.cwi.nl/terracognita2014/terra2014_3.pdf https://github.com/LinkedEOData/GeoTriples
[RD.R2RML]	**R2RML: RDB to RDF Mapping Language** W3C Recommendation, 27/09/2012. http://www.w3.org/TR/r2rml/

ID	Reference
[RD.Strabon]	**Strabon semantic spatiotemporal RDF store** http://www.earthobservatory.eu/Strabon http://www.strabon.di.uoa.gr/ http://www.strabon.di.uoa.gr/userguide http://hg.strabon.di.uoa.gr/Strabon/

1.3 Acronyms and Abbreviations

The following acronyms are used in this manual. It is however not necessary to be familiar with them to make use of the software.

CKAN	Comprehensive Knowledge Archive Network
COTS	Commercial-Off-The-Shelf
EO	Earth Observation
ESA	European Space Agency
FAQ	Frequently Asked Questions
ISO	International Standards Organisation
JRE	Java Runtime Environment
JSON	JavaScript Object Notation
LD	Linked Data
LOD	Linked Open Data
OBEOS	Ontology-Based Earth Observation Products Search
OGC	Open Geospatial Consortium
OS	Operating System
R2RML	RDB to RDF Mapping Language
RDB	Relational Database
RDF	Resource Description Framework
REST	Representational State Transfer
RESTo	REstful Semantic search Tool for geOspatial
RML	RDF Mapping Language
VM	Virtual Machine
XML	eXtensible Markup Language

2 Software Configuration Item Overview

2.1 Description

The Linked Data Prototype is a distributed platform exposing a Web-based user interface that allows users to search and discover Earth Observation and other geo-localized data encoded and published as Linked Data. The search engine communicates with an external component that performs a semantic-based analysis of the user queries.

The platform is powered by an extended version of the CKAN Web access portal. In particular, a CKAN extension has been developed that integrates an ontology browser, adapts the user interface, and extend the default search engine with disambiguation and external communication capabilities.

The integrated ontology browser allows users to select in ontologies concepts related to his needs. The query disambiguation service (also named "query resolver" later on in this document) assigns senses to user search strings and translates these senses (concepts, toponyms, time constraints, named entities, etc.) into low-level search criteria (area and time of interest, keywords, and specific product properties).

Geospatial data is converted into Linked Data using a tool called GeoTriples (see [RD.GeoTriples]), developed by the University of Athens (UoA) in EU FP7 project LEO. GeoTriples takes as input data that reside in plain files and converts them to RDF graphs. The transformation is driven by R2RML mapping definition files. Instructions for converting EO Dataset Series (ISO 19139-2:2012 GMI [ISO−19139-GMI]) and EO Datasets (OGC EOP O&M [OGC−EO-OM]) metadata into Linked Data, compliant with the schema proposed in the "EO Metadata Discovery using Linked Data" discussion paper [EO−LD], are provided in section 3.5.

The Linked Data is ingested in RDF databases (triplestores) powered by Strabon (see [RD.Strabon]), developed by UoA in FP7 project TELEIOS. Strabon exposes a SPARQL interface and supports geo-temporal extensions compatible with GeoSPARQL. Instructions for setting-up a Strabon instance are provided in section 3.6.

2.2 How to Obtain Information

The Linked Data Prototype has been developed in the context of ESA project "Ontology Based Earth Observation Search" (OBEOS). This is described on ESA Research & Service Support (RSS) Web Portal at the following address:

https://wiki.services.eoportal.org/tiki-index.php?page=OBEOS

The software itself is publicly available from the following GitHub repository:

https://github.com/SpaceApplications/ckanext-obeos-ldp

2.3 Software Composition

The OBEOS Linked Data Prototype software is only available in an electronic format. It is not foreseen to distribute it on physical media.

Material necessary to install and run the software is packaged in a single bundle file. This contains the following items:

- A readme file containing material extracted from this document
- Linked Data Prototype Software Configuration File (SCF, this document).
- Linked Data Prototype Software User Manual (SUM, [LDP-SUM]).

- Installation scripts and configuration files.
- A compressed archive containing the code of the CKAN LDP extension.
- R2RML mapping files for EOP O&M metadata.

2.4 Baseline Documents

The following documents, included in the bundle described in section 2.3 apply to the OBEOS Linked Data Prototype software:

- OBEOS Linked Data Prototype Software Configuration File
- OBEOS Linked Data Prototype Software User Manual

2.5 Means to Develop, Modify, Install and Run the Software

Required third party software products are retrieved from public repositories during the (automated or manual) installation of the LDP. These include:

- Apache HTTP Server (automated installation only)
- Nginx HTTP Server (manual installation only)
- CKAN
- Java Runtime Environment (JRE)
- Apache Solr (with embedded Jetty)
- PostgreSQL / PostGIS
- CNES' RESTo Query Resolver and iTag
- UoA' s Strabon
- UoA's GeoTriples
- Memcached

The required version of each of these third-party products is indicated in the manual installation procedure, in section 3.3.

In addition to the above software, the Linked Data Prototype relies on the availability of an external service. This service exposes a SPARQL interface that allows searching and navigating through a number of ontologies. Section 3.7 describes how the Linked Data Prototype must be configured to communicate with the Ontology Service.

2.6 Differences with Previous Versions

This version of the Software Configuration File document describes the initial version of the Linked Data Prototype software.

2.7 Software Status

The development of the Linked Data Prototype has focused on the core capabilities that needed to be demonstrated. The new features are not all compatible with the default behaviour of the CKAN product. This has mostly required the customization of page and snippet templates used to display information in the user interface. Because the CKAN built-in mechanisms still take place, it could be that pre-existing interface components do not behave as expected.

3 Installation and Configuration Instructions

3.1 Introduction

This software installation guide describes two solutions for deploying and configuring the OBEOS Linked Data Prototype software on a target system:

- Using the Linux Containers (LXC) and the Docker technologies for deploying and running the services within sandboxed environments, and
- Manually installing and configuring the services manually in the traditional way.

The two procedures are described independently to each other in sections 3.2 and 3.3, respectively. This guide does not try to propose a combination of both techniques.

Note: The installation of GeoTriples is described in dedicated section 3.5 as this is not an integrated component. It is meant to be used as a stand-alone tool to prepare the data. Also, the optional CNES' query resolver service RESTo is described in dedicated section 3.4.1.

3.2 Container-Based Installation with Docker

Docker is a technology built atop of Linux Containers (LXC) that aims at deploying replicable lightweight environments in a distributed infrastructure. A Docker Container acts as a Virtual Machine, but without the overhead of replicating a complete Operating System (OS).

The remaining of this section describes the installation of the Linked Data Prototype within Docker containers.

3.2.1 Preliminary Requirements

In order to install Docker and build the software, the following requirements shall be met:

- Docker requires a 64-bit version of a recent Linux system. The actual version of the Linux distribution is less relevant than the version of its kernel, which must be 3.10 or higher. The remaining of this manual has been prepared considering an installation in a Linux CentOS 7.x 64 bit distribution. A minimal installation (software selection option available while installation CentOS 7) is sufficient for running the OBEOS LDP containers. Because CentOS is directly derived from Red Hat Enterprise Linux (EL), the instructions should also be applicable to Red Hat EL 7.x.

  The version of the kernel may be obtained using the following command (as any user):

```
$ uname -r
```

- A regular user may build the Docker images and instantiate the containers, provided this user belongs to the **docker** group:

```
$ id -Gn $USER
```

  When not explicitly specified all installation instructions have to be executed using that regular user. Instruction to add a user in the **docker** group (created at the time Docker is installed in the system) is provided in the next section.

- No PostgreSQL server instance should be listening to default port 5432 of the target system as this will be bridged with one of the containers.
- No HTTP server instance should be listening to default port 80 of the target system as this will be bridged with one of the containers.

Note: If a PostgreSQL server and/or an HTTP server are already running on the hosting system and listening to the ports indicated above, alternate ports can be used by the containers, as it will be explained in section 3.2.3, below.

3.2.2 Docker and Docker-Compose Installation

Docker-Compose is a tool that facilitates the definition and the management of multi-container Docker applications.

Installation steps:

1. The procedure for installing the Docker tools on a CentOS 7.x distribution is described in the follow guide:

   https://docs.docker.com/engine/installation/centos/

2. As root user, add the regular user to the **`docker`** group:

```
# usermod -aG docker $USER
```

3. As root user, execute the following command to install the Docker-Compose tool:

```
# curl -L \
https://github.com/docker/compose/releases/download/1.5.2/docker-compose-
`uname -s`-`uname -m` > /usr/local/bin/docker-compose
```

3.2.3 Bundle File Extraction

The OBEOS Linked Data Prototype installation files are provided in a single compressed archive named **`OBEOS-LDP-InstallBundle-<date>.tgz`**. The archive may be extracted in the location of your choice as follows:

```
$ cd <your-location>
$ gzip -dc <obeos-ldp-install-bundle-file> | tar xvf -
```

At that point, the sub-folder **./ldp-install/docker.centos7** shall contain the following files and folders:

```
./ldp-install/docker.centos7/
├── build.All.sh
├── build.CKANExtOnly.sh
├── common_shared_folder/
├── docker/
├── docker-compose.yml
├── Dockerfile
├── _etc/
├── _solr/
└── _src/
```

Should either PostgreSQL or the HTTP server need to be configured to connect to an alternate port, adjust the port numbers in the following file:

```
./ldp-install/docker.centos7/docker-compose.yml
```

3.2.4 Configure the Portal URL

The deployment of the OBEOS Linked Data Prototype requires the LDP Portal base URL to be configured in the container running CKAN.

To configure the site URL, edit the following file:

```
./ldp-install/docker.centos7/Dockerfile
```

And indicate the appropriate value as follows:

```
ENV SITE_URL domain.example.com
```

This value will be automatically replicated in the file **`/etc/ckan/default/ckan.ini`** located in the CKAN container.

3.2.5 Build the Container Images

Docker containers are based on Docker Container Images, just as Virtual Machine instances are based on Virtual Machine images. Docker gives the possibility to create Docker Container Images from "**`Dockerfile`**" files which act as installation scripts.

When a system uses different services, a set of various Docker images and their interdependencies can be described in a **`docker-compose`** file. This file allows building and instantiating the images, and controlling the instances

The Linked Data Prototype is made of a number of services. The Docker-based installation procedure will deploy them in separate images. This allows distributing the services on different hosts, if necessary.

The Docker Container images listed in Table 2 will be created. The first two images are base images that are used to build the next ones. They are not instantiated and executed as containers. The image sizes indicated in the table are approximations.

Table 2 – Images and Containers Description and Size

Image Name	~ Size	Container Name	Description
centos	197 MB	N/A	Base CentOS7 image pulled from Docker Hub
dockercentos7_centos-base	317 MB	N/A	Base container for every other containers
dockercentos7_ckan	883 MB	dockercentos7_ckan_1	Runs Apache HTTP Server, Python, CKAN and the OBEOS Linked Data Prototype extension.
dockercentos7_solr	898 MB	dockercentos7_solr_1	Runs Java Runtime Environment (JRE), Jetty and Apache Solr. Solr is used by CKAN to catalogue and index its data.
dockercentos7_postgres	625 MB	dockercentos7_postgres_1	Runs PostgreSQL database engine.
dockercentos7_data	317 MB	dockercentos7_data_1	Contains the dynamic data of the platform including the PostgreSQL databases.
dockercentos7_strabon-base	634 MB	dockercentos7_strabon_ontoservice_1	Runs a Strabon instance populated with ontologies and mapping links.
		dockercentos7_strabon_landsat8_1	Runs a Strabon instance populated with Landsat-8 L1T and L1gT products metadata.
~ TOTAL	3871 MB		

CKAN, Solr and the Strabon containers (image instances) connect to the same PostgreSQL engine (running in the "_postgres_1" container) to store and access their data. The actual PostgreSQL databases are however located in the "_data_1" container.

To be on the safe size, the disk space available before starting the installation must be at least the double of the total size of the Docker images. It is thus recommended to have at least 8 GB of free disk space. During the operations, only the "_data_1" container will increase in size, as it is the only container used to store new data.

Tests have shown that 4 GB of RAM and a CPU with 4 cores are sufficient to run all the containers deployed on the same host.

The Docker Container images are built on top of base images that must be generated first.

Enter the **`./ldp-install/docker.centos7`** folder to build the CentOS Base image and the Strabon base image using the following command:

```
$ docker-compose build --no-cache centos-base strabon-base
```

Then, build the software specific images by executing the following command:

```
$ docker-compose build --no-cache
```

The traces of the build process shall end with:

```
Successfully built
```

3.2.6 Start the Linked Data Prototype Software

To start all the Docker instances in a single operation, enter the **./ldp-install/docker.centos7** folder and execute the following command:

```
$ docker-compose up
```

This shall print the following traces in the terminal:

```
Starting dockercentos7_data_1...
Starting dockercentos7_solr_1...
Starting dockercentos7_postgres_1...
Starting dockercentos7_strabon_ontoservice_1...
Starting dockercentos7_strabon_landsat8_1...
Starting dockercentos7_ckan_1...
// Continuation of the detailed logs for each service
```

Upon the first execution the container instances do not exist and are created by Docker. This operation requires a couple of minutes to complete. Subsequent executions are almost instantaneously.

A process manager such as systemd[1] or supervisord[2] may be used to automatically start the containers at system start-up.

The containers internal ports are bound to the host ports according to the following table:

Table 3 – Containers Services and Ports Mapping

Host Port	Container Name	Container Port	Service
80	dockercentos7_ckan_1	80/tcp	Apache
5000		5000/tcp	Paster
2222		22/tcp	SSH
8983	dockercentos7_solr_1	8983/tcp	Solr
5432	dockercentos7_postgres_1	5432/tcp	Postgres
8080	dockercentos7_strabon_ontoservice_1	8080	Strabon
8081	dockercentos7_strabon_landsat8_1	8080	Strabon
-	dockercentos7_data_1	-	-

The OBEOS Linked Data Prototype end-user interface is accessible using a traditional Web browser at the following address: http://localhost

[1] http://freedesktop.org/wiki/Software/systemd/
[2] http://supervisord.org/

3.2.7 Stop the Linked Data Prototype Software

To stop the images, enter the `./ldp-install/docker.centos7` folder and execute the following command. It must not be done to pursue the installation of the LDP.

```
$ docker-compose stop
```

This shall print the following traces in the terminal:

```
Stopping dockercentos7_ckan_1... done
Stopping dockercentos7_strabon_ontoservice_1... done
Stopping dockercentos7_strabon_landsat8_1... done
Stopping dockercentos7_postgres_1... done
Stopping dockercentos7_solr_1... done
Stopping dockercentos7_data_1... done
```

3.2.8 Add a SysAdmin User

CKAN administration requires a SysAdmin user to be created. This is not performed by default.

The following command creates a new user with administration rights in the CKAN portal. The command must be executed in the host system as it will connect automatically to the indicated (**`dockercentos7_ckan_1`**) container. The containers must be started.

The actual user short name may be freely chosen by replacing the parameter **`<admin_short_name>`** (e.g. "admin"):

```
$ docker exec -ti dockercentos7_ckan_1 \
/usr/lib/ckan/default/bin/paster --plugin=ckan sysadmin \
add <admin_short_name> --config=/etc/ckan/default/ckan.ini
```

When prompted, enter a password - this is the password you will use to log in to CKAN.

If the container has been started from a different folder, it can be necessary to adapt the container name. If so, you may use the following command to obtain the name of the containers present in the system:

```
$ docker ps
```

3.2.9 Troubleshooting

When the images are built with "`docker-compose build --no-cache`", Docker complains that URLs cannot be fetched.

This can happen when the system uses DNS servers that are filtered by the network infrastructure. In order to be able to fetch all the dependencies, add the Google DNS addresses into the configuration file **`/etc/resolv.conf`**:

```
nameserver 8.8.8.8
nameserver 8.8.4.4
```

When the services are started with "`docker-compose up`", they start and quit automatically shortly after.

Some deprecated containers are being used. To clean them up and restart with a fresh installation execute the following commands:

```
# Stop the current containers
docker-compose stop
# Remove all the stopped containers.
# Be carefull, this will remove all the stopped containers from the
system, even those used by other services.
docker rm $(docker ps -q -a);
# Remove all untagged containers
sudo docker rmi $(sudo docker images -f "dangling=true" -q)
# Rebuild the system
docker-compose build --no-cache
# Run again
docker-compose up
```

When `docker-compose` is executed, the error "`Couldn't connect to Docker daemon at http+unix://var/run/docker.sock - is it running?`" is thrown.

The **`docker`** group shall be added to the user that executes **`docker-compose`**:

```
usermod -aG docker ${USER}
```

To transfer files between containers, use the folder of the host system shared with all the containers.

The shared folder is located in the host at:

```
./ldp-install/docker.centos7/common_shared_folder/
```

And in each container at:

```
/shared_folder
```

When a file is dropped in this folder, from the host or from any of the containers, it is immediately visible for the others.

The log traces produced by CKAN and its extensions (including the LDP plugin) may be found in the CKAN container, in the following two files:

```
/var/log/httpd/ckan_default.custom.log
/var/log/httpd/ckan_default.error.log
```

For accessing these files, one must open a shell in the CKAN container by issuing the following command:

```
$ docker exec -ti dockercentos7_ckan_1 /bin/bash
```

Type "**`exit`**" or hit "Ctrl-D" to close the shell and exit the CKAN container.

To re-start the services running in a container (e.g. Apache in the CKAN container), start a shell in the container and issue the following command:

```
$ sv restart supervisord
```

3.3 Manual Installation Procedure

This section details the procedure to manually install all the components of the OBEOS Linked Data Prototype and their dependencies on a target system.

In particular, the installation of the following software is described:

- PostgreSQL
- Java Runtime Environment
- Apache Solr
- CKAN
- Apache HTTP Server
- Linked Data Prototype extension to CKAN

The procedure targets a Linux CentOS 7.X 64 bit distribution. This distribution can be downloaded at: https://www.centos.org/download/.

3.3.1 Linked Data Prototype Bundle

The OBEOS Linked Data Prototype installation files are provided in a single compressed archive named `OBEOS-LDP-InstallBundle-<date>.tgz`. The archive may be extracted in the location of your choice as follows:

```
$ cd <your-location>
$ gzip -dc <obeos-ldp-install-bundle-file> | tar xvf -
```

At that point, the sub-folder **./ldp-install/manual.centos7** shall contain the files required by the manual installation procedure of the LDP.

3.3.1.1 Temporary Installation Environment Variables

Environment variables are needed for the duration of the installation process. The variables can be set directly in the terminal, or persisted in the file `~/.bashrc`. In the latter case, it is recommended to remove them once the installation as this is confidential information.

The installation environment variables are detailed in the following listing. The name and passwords shall be changed for security reasons.

```
# Default credentials if none have been provided at runtime.
# You should at least set secure passwords.
export CKAN_DBPG=ckan
export CKAN_USER=ckan_user
export CKAN_PASS=ckan_pass

export DATASTORE_DB=datastore
export DATASTORE_USER=datastore_user
export DATASTORE_PASS=datastore_pass

export PGDATA=/var/lib/postgresql/9.3/main
export PGMAIN=/etc/postgresql/9.3/main

# Folder of the Linked Data Prototype bundle files in folder
"manual_install".
export LDP_MANUALINSTALL=/path/to/the/bundle/manual_install
```

3.3.1.2 Administration Mode

All the steps in this installation procedure must be performed as an root user.

Environment variables are sometimes used in the commands however, there may have been defined by a normal user. In order to access these variables while running commands as root user, it is therefore recommended to log in as an administrator with the **--preserve-environment** option:

```
$ su --preserve-environment
```

3.3.2 PostgreSQL Installation and Configuration

PostgreSQL[3] is an object-relational database management system (ORDBMS) with an emphasis on extensibility and standards compliance.

Software Packages

Name	Version	Description
postgresql	9.2	PostgreSQL client programs
postgresql-server	9.2	The programs needed to create and run a PostgreSQL server
postgresql-contrib	9.2.15	Extension modules distributed with PostgreSQL
postgis	2.0.7	Geographic Information Systems Extensions to PostgreSQL
inotify-tools	3.14	Command line utilities for inotify

PostgreSQL Installation

Install the required packages as follows:

```
yum -y update

# Install PostgreSQL and PostGIS.
yum -y install \
        postgresql \
        postgresql-server \
        postgresql-contrib \
        postgis \
        inotify-tools
```

Once the environment variables are set, execute the following commands:

```
mkdir -p $PGMAIN

# Default credentials if none have been provided at runtime.
mkdir -p $PGDATA && chown -R postgres $PGDATA && chmod -R 700 $PGDATA

# Copy setuser util in sbin for use in configureDB script
cp $LDP_MANUALINSTALL/utils/setuser /sbin/

# Create the PostGIS template and initialize the CKAN database
. $LDP_MANUALINSTALL/services/postgres/configureDB
```

[3] http://postgresql.org/

```
# Start Postgreql Service
# -> The Service is automatically started by the configureDB script
```

3.3.3 Apache Solr Installation and Configuration

Apache Solr[4] is an open source enterprise search platform, written in Java, from the Apache Lucene[5] project.

Software Packages

Name	Version	Description
java-1.7.0-openjdk-headless	1.7.0	OpenJDK runtime environment without audio and video support
Solr	4.10.1	Enterprise search server based on Lucene

Apache Solr Installation

Install the required packages as follows:

```
yum -y update

# Install Java
yum -y install \
       java-1.7.0-openjdk-headless
```

In the current shell or in **`~/.bashrc`**, define the following environment variables:

```
export SOLR_HOME=/opt/solr/ckan/solr
export SOLR_VERSION=4.10.1
export SOLR=solr-$SOLR_VERSION
```

Once the environment variables are set, execute the following commands:

```
# Install Solr
mkdir -p /opt/solr
curl -O \
   https://archive.apache.org/dist/lucene/solr/$SOLR_VERSION/$SOLR.tgz
tar zxf $SOLR.tgz -C /opt/solr --strip-components 1

# Create the CKAN collection
mv /opt/solr/example /opt/solr/ckan
mv /opt/solr/ckan/solr/collection1 $SOLR_HOME/ckan

echo name=ckan > $SOLR_HOME/ckan/core.properties

# Copy the CKAN schema
cp $LDP_MANUALINSTALL/services/solr/schema.xml \
   $SOLR_HOME/ckan/conf/schema.xml

# Start Solr
cd /opt/solr/ckan
java -jar /opt/solr/ckan/start.jar &

# Remove Solr archive
rm -rf $SOLR.tgz
```

[4] http://lucene.apache.org/solr/
[5] http://lucene.apache.org/

Once the installation is done, Solr administration interface shall be accessible through:

```
http://localhost:8983/solr/admin
```

An empty **ckan** collection is accessible in a Web browser at the following address:

```
http://localhost:8983/solr/#/ckan
```

3.3.4 CKAN Installation and Configuration

CKAN[6] is an Open Source Data Portal. This software relies on PostgreSQL to store its data and Apache Solr to index and perform the searches inside the data.

Software Packages

Name	Version	Description
gcc	4.8	Various compilers (C, C++, Objective-C, Java, ...)
python-devel	2.7	The libraries and header files needed for Python development
python-pip	7.1	A tool for installing and managing Python packages
python-virtualenv	1.10	Tool to create isolated Python environments
postgresql-devel	9.2	PostgreSQL development header files and libraries
libxml2-devel	2.9	Libraries, includes, etc. to develop XML and HTML applications
libxslt-devel	1.1	Development files for libxslt
memcached	1.4	High Performance, Distributed Memory Object Cache
libmemcached	1.0	Client library and command line tools for memcached server
libmemcached-devel	1.0	Header files and development libraries for libmemcached
git	1.8	Fast Version Control System

CKAN Installation

Install the required packages as follows:

```
yum -y update

# Install required packages
yum -y install \
       gcc \
       python-devel \
       python-pip \
       python-virtualenv \
       postgresql-devel \
       libxml2-devel \
       libxslt-devel \
       memcached \
       libmemcached \
```

[6] http://ckan.org

```
        libmemcached-devel \
        git
```

In the current shell or in **~/.bashrc**, define the following environment variables:

```
# Create Environment variables
export CKAN_HOME=/usr/lib/ckan/default
export CKAN_CONFIG=/etc/ckan/default
export CONFIG_FILE=ckan.ini
export CONFIG_OPTIONS=custom_options.ini
export CKAN_DATA=/var/lib/ckan
export CKAN_INI=$CKAN_CONFIG/$CONFIG_FILE
export CKAN_SYS_ADMIN=<admin_short_name>
export SITE_URL=<domain.example.com>
export PYTHON_EGGS_CACHE=/usr/share/httpd/.python-eggs
```

Notes:

- **<admin_short_name>** is the login name of the SysAdmin user to be created in the CKAN Portal. This may be freely chosen (e.g. "admin").
- **<domain.example.com>** is the CKAN Portal base URL.

Once the environment variables are set, execute the following commands:

```
# Create directories & virtual env for CKAN
virtualenv $CKAN_HOME
mkdir -p $CKAN_CONFIG $CKAN_DATA /var/log/ckan

# copy CKAN and any extensions in the source directory
git clone https://github.com/ckan/ckan.git $CKAN_HOME/src/ckan
$CKAN_HOME/bin/pip install pip==1.4.1

# install any extensions
$LDP_MANUALINSTALL/services/ckan/pip_install_req.sh
ln -s $CKAN_HOME/src/ckan/ckan/config/who.ini $CKAN_CONFIG/who.ini

# Make config file
$CKAN_HOME/bin/paster make-config ckan ${CKAN_CONFIG}/${CONFIG_FILE}

# Configure the site URL
sed -e "s/#ckan.site_url/ckan.site_url/g" $CKAN_INI
sed -i -e \
   '/ckan.site_url[[:space:]]\+=[[:space:]]\+/ s@= .*@= '"$SITE_URL"'@' \
   $CKAN_INI

# Add a new CKAN sysadmin user
$CKAN_HOME/bin/paster --plugin=ckan sysadmin add $CKAN_SYS_ADMIN \
       --config=$CKAN_INI
```

At this point Solr and PostgreSQL are configured and running.

3.3.5 Nginx HTTP Server Installation and Configuration

An Nginx HTTP Server[7] is installed in order to serve the pages of CKAN. The Nginx server is configured as a revers proxy on the localhost port 5000 serving CKAN locally.

[7] http://nginx.org/

Software Packages

Name	Version	Description
nginx	1.6	A high performance web server and reverse proxy server

Nginx HTTP Server Installation

Install the required packages as follows:

```
yum -y update

# Install required package
yum -y install \
       nginx
```

Execute the following commands to configure the server:

```
# Give permissions on CKAN folder
chmod -R 755 $CKAN_DATA

# Configure nginx
cp $LDP_MANUALINSTALL/etc/nginx/ckan.conf /etc/nginx/conf.d/
systemctl restart nginx

# Start nginx on boot
systemctl enable nginx
```

3.3.6 CKAN Extension Installation and Configuration

The CKAN extension contains the code for the Linked Data Prototype software. It is installed through a standard **setup.py** operation.

Please note that once the installation is done, the CKAN Extension sources shall remain in the folder from where **setup.py** was triggered since some references to this folder are being created through the installation.

CKAN Extension Installation

Execute the following commands to install the CKAN extension:

```
# Configure OBEOS CKAN extensions
tar xzvf $LDP_MANUALINSTALL/src/ckanext-obeos.tar.gz  -C $CKAN_HOME/src/
cd $CKAN_HOME/src/ckanext-obeos/
$CKAN_HOME/bin/python setup.py develop
mkdir -p /var/log/memcached/
mkdir -p /var/www/.python-eggs

# Copy any custom config
cp $LDP_MANUALINSTALL/services/ckan/custom_options.ini $CKAN_CONFIG/

# Install the javascript files
cd ~ && \
  curl -O http://nodejs.org/dist/v0.10.30/node-v0.10.30-linux-x64.tar.gz
tar --strip-components 1 -xzvf ~/node-v* -C /usr/local
git config --global url."https://".insteadOf git://
npm install -g bower
cd $CKAN_HOME/src/ckanext-obeos/ckanext/rdfview/public/static && \
  bower install --allow-root --config.interactive=false
```

Start CKAN Server and visit http://localhost:5000 to check if everything if fine:

```
# Start the CKAN Server
$LDP_HOME/bin/paster serve /etc/ckan/default/ckan.ini
```

You can now configure CKAN by executing the following script:

```
# Configure the options in /etc/ckan/default/custom_options.ini
$LDP_MANUALINSTALL/services/ckan/configureCKAN.sh
```

3.3.7 Supervisor Installation and Configuration

Supervisor[8] is a Process Control System. It is a client/server system that allows its users to remotely monitor and control processes on UNIX-like operating systems.

Software Packages

Name	Version	Description
supervisor	3.1	A System for Allowing the Control of Process State on UNIX

Supervisor Installation

A Supervisor configuration file for the Linked Data Prototype is available in the bundle and can be installed as follow:

```
yum install -y supervisor
mkdir -p /etc/supervisor/conf.d/
# Create folders for logs
mkdir -p /var/log/solr/
mkdir -p /var/log/postgres/
# Copy configuration file
cp $LDP_MANUALINSTALL/etc/supervisor/supervisord.ini /etc/supervisord.d/
# Start supervisor Daemon
supervisord

# Start supervisor on boot
systemctl enable supervisord
```

With Supervisor services can be started, stopped and restarted on demand.

For instance, to restart Solr, issue the following command:

```
$ supervisorctl restart solr
```

3.3.8 Starting and Stopping the Linked Data Prototype Software

To start the Linked Data Prototype, run the following command:

```
$ systemctl start supervisord nginx
```

To stop the Linked Data Prototype, run the following command:

```
$ systemctl stop supervisord nginx
```

[8] http://supervisord.org/

3.3.9 Troubleshooting

The following command may be used to verify that all services are up and running:

```
$ systemctl list-units
```

The log traces produced by CKAN and its extensions (including the LDP plugin) may be found in the following file:

```
/var/log/supervisor/ckan-stdout.log
```

3.4 Query Resolver Service

The Linked Data Prototype uses a Query Resolver service for processing the free-text search queries entered by the user and identify known entities. The LDP is capable to interact with such a service called RESTo, developed by CNES (see [RD.RESTo]), or with a service accessible on-line, called Query Analyzer, developed by Space Applications Services.

To use the pre-deployed Query Analyzer, the service endpoint URL must be configured in the CKAN LDP plugin, as described in section 3.7.1, page 25. This option removes the need to manually install the RESTo software.

Otherwise, the instructions for manually downloading and installing the OpenSource Software RESTo are provided hereafter.

3.4.1 Installation of CNES' RESTo OpenSource Software

RESTo is an EO product search engine that exposes an OpenSearch interface. It also exposes a URL that allows disambiguating a user query expressed in plain text. In particular, it is capable of recognizing different types of tokens such as missions, instruments, toponyms, time expressions and a pre-defined list of keywords. In the context of the Linked Data Prototype, only this latter capability is used.

Instructions for installing and setting-up RESTo are available on its home page on GitHub: https://github.com/jjrom/resto.

Notes:

- The RESTo installation instructions propose to install the iTag server [RD.iTag]. In order to enable the toponyms detection feature, this must be done.
- Because the RESTo search engine is not used in the LDP, the installation step where collections metadata are ingested in the RESTo database may be skipped.

For testing purpose, the query resolution service of RESTo may be invoked as follows:

> http://<domain>:<port>/2.1/api/query/analyze.json?_pretty=true&lang=en&q=Pleiades%20images%20of%20Toulouse%20in%20april%202015

The response must be a JSON document that repeats the query string, and provides "what", "where" and "when" clauses derived from it.

Example (shortened) response:

```
{
    "query": "Pleiades images of Toulouse in april 2015",
    "language": "en",
    "analyze": {
        "What": {
            "eo:platform": "PHR1A|PHR1B"
        },
        "When": {
            "times": [
                {
                    "time:start": "2015-04-01T00:00:00Z",
                    "time:end": "2015-04-30T23:59:59Z"
                }
            ]
        },
        "Where": [
            {
                "name": "Toulouse",
                "type": "toponym",
                "country": "France",
                "geo:lon": 1.44367,
                "geo:lat": 43.60426,
                "ccode": "FR",
                "fcode": "PPLA",
                "admin1": "B3",
                "admin2": "31",
                "population": 433055,
                "elevation": 0,
                "gtopo30": "150",
                "timezone": "Europe\/Paris",
                "SeeAlso": [ removed content ]
            }
        ],
        "Explained": [ removed content ]
    }
}
```

3.5 Linked Data Generation with GeoTriples

GeoTriples is an open-source software product developed by the University of Athens (see [RD.GeoTriples]). The tool includes an RML processor, a superset of R2RML (see [RD.R2RML]). Mappings that express the transformation of source data to Linked Data are written in the RML language. An overview diagram is provided in Figure 1, below.

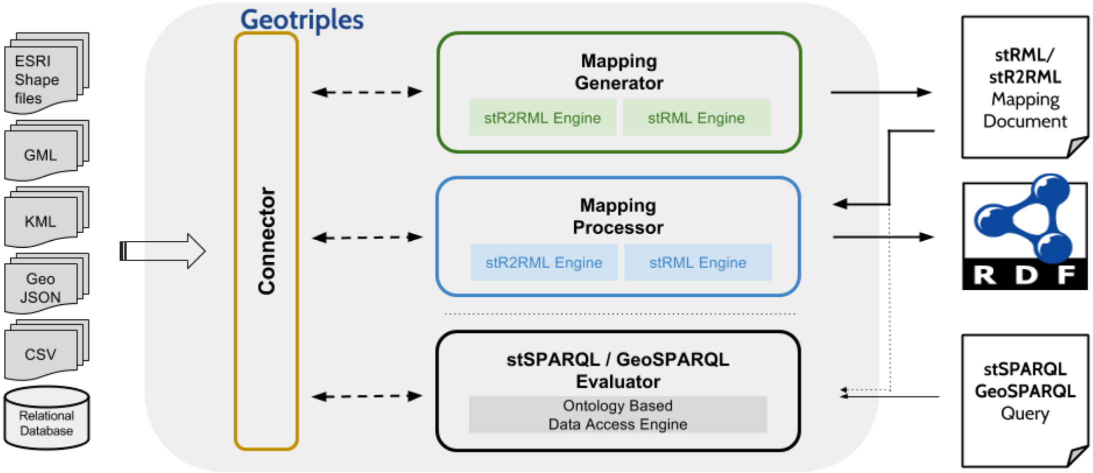


Figure 1 – GeoTriples Overview Diagram[9]

The transformation to Linked Data and the ingestion into the RDF Database is an offline process that must be run manually each time new or updated data must be made available as Linked Data.

[9] Source: [RD.GeoTriples]

The Linked Data Prototype data package described in section 2.3 includes R2RML files that allow converting datasets encoded in OGC EOP O&M (OGC 10-157) into Linked Data compliant with the schema proposed in the "*EO Metadata Discovery using Linked Data*" discussion paper [EO-LD].

GeoTriples Installation

A pre-compiled version of GeoTriples is available on the GeoTriples home page on GitHub: http://linkedeodata.github.io/GeoTriples/. Click on the "Command Line Tool - Jar Download" button located in the page banner to download the executable Jar (Java archive) file.

At the time of preparing the current document, the direct URL is the following:

http://linkedeodata.github.io/GeoTriples/geotriples-1.0.5-SNAPSHOT-cmd.jar

The JAR file must be executable to allow one to use it as a command line tool:

```
$ chmod ugo+x geotriples-1.0.5-SNAPSHOT-cmd.jar
```

It may also be convenient to create a symbolic link to shorten the command name:

```
$ ln -s geotriples-1.0.5-SNAPSHOT-cmd.jar geotriples-cmd
```

Linked Data Generation

The following files are provided in the **/ldp-mapping** folder of the LDP software bundle:

```
ldp-mapping/
├── convert_dataset.sh
├── convert_dataset_series.sh
├── dataset_namespaces.ns
├── dataset_series_namespaces.ns
├── OGC-EO-Coll-2-RDF-mapping.rml.ttl
└── OGC-EOP-OM-2-RDF-mapping.rml.ttl
```

The two "**OGC-EO***" files implement the conversion from the ISO and OGC standard formats to Linked Data as described in the OGC Discussion Paper "EO Metadata Discovery using Linked Data" [EO-LD]. The two "**ns**" files provide the list of namespaces to be included in the output files. Finally, the two helper scripts ("**sh**" files) may be executed to convert the EO dataset series and datasets metadata.

To convert ISO dataset series metadata located in file "**input_file.xml**", and store the result in file "**output_file.rdf**", enter the **ldp-mapping** folder and execute the following command:

```
$ ./convert_dataset_series.sh <input_file.xml> <output_file.rdf>
```

To convert OGC EOP O&M dataset metadata located in file "**input_file.xml**", and store the result in file "**output_file.rdf**" execute the following command:

```
$ ./convert_dataset.sh <input_file.xml> <output_file.rdf>
```

The following paragraphs explain how the same results may be obtained bypassing the helper scripts and using GeoTriples directly.

Linked Data Generation Explained

The data source used for a specific mapping is specified in the corresponding R2RML file itself. It thus means it is necessary to edit the provided mapping file to indicate which local file contains the data to be converted.

In order to process an input file encoded in XML using a pre-existing mapping file, the following command must be used:

```
$ geotriples-cmd dump_rdf [-ns nsfile] [-f format] [-b baseURI] \
         [-o rdfoutfile] [-s epsgcode] -rml inputRMLmappingfile
```

The parameters are as follows:

- **dump_rdf**: tells GeoTriples that it must generate an RDF document.
- **nsfile** (optional by default but required in the LDP context): file containing the namespaces to be prepended to the output file.
- **format** (optional): N-TRIPLE (default), RDF/XML, RDF/XML-ABBREV, or TURTLE.
- **baseURI** (optional): base URI for RDF output.
- **rdfoutfile** (optional): output file name (default: stdout).
- **epsgcode** (optional): projection to be used while generating the RDF output.
- **-rml**: use the RML processor (for XML, JSON, and CSV files).
- **inputRMLmappingfile**: RML file containing the mapping definition.

For example, in order to convert the file "**ogc-eop-om-dataset.xml**" which contains EOP O&M compliant metadata, the following steps must be performed:

- Edit the R2RML mapping file "**OGC-EOP-OM-2-RDF-mapping.rml.ttl**" (encoded in RDF Turtle) and enter the path and the name of the input file in the "**rml:logicalSource**" / "**rml:source**" property as follows:

```
rml:logicalSource [
   rml:source "path/to/ogc-eop-om-dataset.xml";
   rml:referenceFormulation ql:XPath;
   rml:iterator "/opt:EarthObservation"];
```

- Execute GeoTriples using the following command, asking the generated Linked Data to be stored in the new file "**ogc-eop-om-dataset.rdf**":

```
$ geotriples-cmd  dump_rdf -ns dataset_namespaces.ns \
                  -o ogc-eop-om-dataset.rdf \
                  -rml OGC-EOP-OM-2-RDF-mapping.rml.ttl
```

The generated RDF document is stored in the specified output file. The file is overwritten if it already existed. This RDF document may then be ingested in the Strabon RDF database, as described in the next section.

Similarly, for converting dataset series metadata, the above procedure must be executed using the **OGC-EO-Coll-2-RDF-mapping.rml.ttl** file as follows:

```
$ geotriples-cmd  dump-rdf -ns dataset_series_namespaces.ns \
                  -o ogc-eop-dataset-series.rdf \
                  -rml OGC-EOP-Coll-2-RDF-mapping.rml.ttl
```

3.6 Linked Data Storage with Strabon

Strabon is OpenSource software developed by the University of Athens (see [RD.Strabon]). The source code is available from UoA's Mercurial repository: http://hg.strabon.di.uoa.gr/Strabon/.

The container-based installation procedure described in section 3.2 includes the installation and initialisation of the Strabon instances necessary to start using the Linked Data Prototype.

The instructions provided hereafter must be followed to manually install the Linked Data Prototype, or to deploy additional Strabon instances.

Strabon Installation

Instructions for installing Strabon on an Ubuntu (Debian-based) system are available on the Strabon wiki: http://www.strabon.di.uoa.gr/userguide. Transposing the instructions to the CentOS (Red Hat-based) system mainly consists in using the **`yum`** package manager instead of **`apt-get`** to install software packages.

Strabon stores its data in a PostgreSQL, PostGIS-enabled database. This software has already been installed in section 3.3.2.

To create the database **`<dbname>`** of a given Strabon instance, issue the following command (as root user):

```
# setuser postgres createdb <dbname> -T template_postgis
```

Strabon is a Web application meant to be run in an Apache Tomcat 7 instance.

Install the Java Runtime Environment (JRE) is not already done (it is also required by Apache Solr, see section 3.3.3):

```
# Install Java
yum -y install \
       java-1.7.0-openjdk-headless
```

Install Apache Tomcat, if not already present:

```
wget http://archive.apache.org/dist/tomcat/tomcat-7/v7.0.68/bin/apache-tomcat-7.0.68.tar.gz && \
       tar -xvf apache-tomcat-7.0.68.tar.gz && \
       rm apache-tomcat*.tar.gz && \
       mv apache-tomcat* /opt/tomcat
```

A pre-compiled version of Strabon may be found at the following address:

http://maven.strabon.di.uoa.gr/service/local/repositories/releases/content/eu/earthobservatory/strabon-endpoint/3.3.1/strabon-endpoint-3.3.1.war. Download and decompress the Strabon WAR file in the Apache Tomcat "**`webapps`**" folder:

```
# cd /tmp/
# wget \
http://maven.strabon.di.uoa.gr/service/local/repositories/releases/content/eu/earthobservatory/strabon-endpoint/3.3.1/strabon-endpoint-3.3.1.war
# setuser tomcat mkdir /opt/tomcat/webapps/strabon
# setuser tomcat unzip strabon-endpoint-3.3.1.war \
                      -d /opt/tomcat/webapps/strabon
```

Note: the above command decompresses the WAR content under the folder named "**`strabon`**". The actual Strabon context thus becomes "**`strabon`**". Multiple instances of Strabon may be deployed in the same Tomcat installation provided they are given different names and are connected to different databases.

Enter the folder **`/var/lib/tomcat7/webapps/strabon/WEB-INF`** and edit the file **`connection.properties`** to configure the information needed by Strabon to connect to its database. In particular, the hostname, the port (default: 5432), the username and password, as well as the database name must be specified.

When this is done, Apache Tomcat may be started (as any user as the **`systemctl`** command line tool will ask for the identity to be used):

```
# systemctl start tomcat
```

The Strabon Web-based client is available at the following address:

http://<localhost>:8080/strabon/

Tomcat may be restarted or fully stopped by issuing the following commands:

```
# systemctl restart tomcat
# systemctl stop tomcat
```

Linked Data Storage

Linked Data may be inserted in the datastore in several manners. The choice is mainly dependent on the volume of the data to be inserted.

- Small documents (e.g. a few kilobytes) may be processed directly using the Strabon endpoint WebUI. To do so, navigate to the Strabon endpoint and select Explore/Modify operations the Store, in the left menu. In the Store Page (see Figure 2, below), paste the document content into the "Direct Input" field, select the appropriate RDF Format, then click on the Store Input button.

  If you are not yet authenticated, a login form will be displayed. The credentials are the ones stored in the **`credentials.properties`** file located in the **`WEB-INF`** folder of the web application.

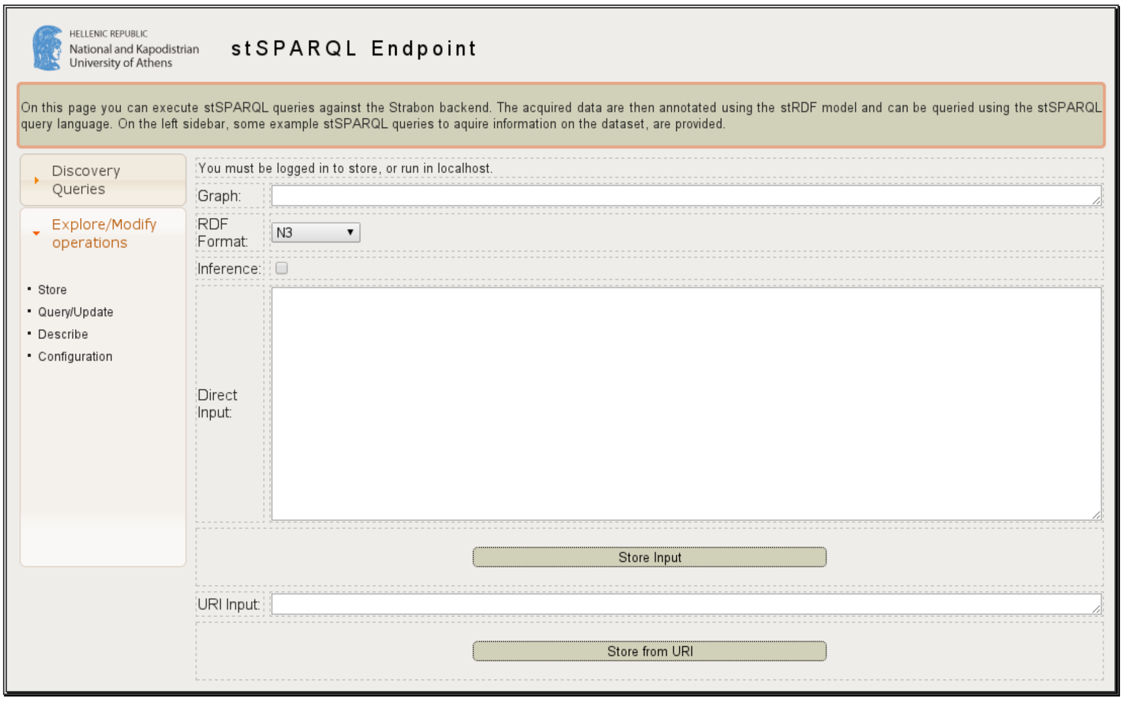
- Bigger documents may be processed using the same page using the "URI Input" field. If the document is available on-line, its URL may be inserted in the field. If not, it must be copied to the Strabon host, then a "**`file:/<path>/<filename>`**" URL that includes the absolute path to the file must be provided in the field.

  Select the appropriate RDF Format, then click on the Store from URI button.

- The Strabon package comes with a **`strabon`** script (located in the **`scripts`** folder) that may be used to insert data from the command line:

  ```
  ./strabon -db dbname -p port -h hostname \
          -u username -pass password store <file.nt>
  ```

  The advantage with this approach over the Strabon WebUI is that it is more configurable with respect to memory usage. In particular, it is possible to pass the option **`-M`** before the **`store`** command to use a pre-configured amount of memory: 1.5GB, or specify a value using the option **`–MM MULTI`**, where **`MULTI`** is a multiplier for 1GB.



On this page you can execute stSPARQL queries against the Strabon backend. The acquired data are then annotated using the stRDF model and can be queried using the stSPARQL query language. On the left sidebar, some example stSPARQL queries to aquire information on the dataset, are provided.

You must be logged in to store, or run in localhost.

Figure 2 – Strabon Linked Data Store Page

3.7 Linked Data Prototype Configuration

This section describes the procedure to configure the Linked Data Prototype software, and to register new Strabon and FedEO endpoints.

Should CKAN be installed in a Docker container, one must open a shell prompt in that container before proceeding with the configuration procedure. To connect to the CKAN container, issue the following command:

```
$ docker exec -ti dockercentos7_ckan_1 /bin/bash
```

When done with the configuration, type "**exit**" or hit "Ctrl-D" to close the connection with the container.

The configuration properties are located in the **obeos_config.json** file located in the following folder:

```
/usr/lib/ckan/default/src/ckanext-obeos/ckanext/obeos/
```

Changes in the configuration file are only taken into account after a restart of the Linked Data Prototype software (see sections 3.2.6 and 3.3.8).

The following sections describe the available configuration options.

3.7.1 Configuration of Remote Services

The Linked Data Prototype CKAN extension communicates with remote services to perform its tasks. These include the Ontology Service, used by the Cross-Ontology Browser, and the Query Resolver, used to analyse the search strings entered by the users.

The URL of the two services must be indicated in the configuration file as shown in the following example:

```
{
  "QUERY_RESOLVER_URL":
    "http://obeos.spaceapplications.com/qa-resto/rest/2.1/api/query/analyze.json",
  "ONTOLOGY_SERVICE_URL":
    "http://obeos.spaceapplications.com/ontologies/",
  "ENDPOINTS": [
  ]
}
```

Note: The Query Resolver endpoint URL included in the above example refers to the service implemented and hosted by Space Applications Services (example invocation: http://obeos.spaceapplications.com/qa-resto/rest/2.1/api/query/analyze.json?q=landsat8).

To use CNES' RESTo service (whose installation instructions are provided in section 3.4.1), the URL of the RESTo instance must be indicated instead.

3.7.2 Registration of a Strabon Endpoint

A new Strabon endpoint may be configured by editing the **obeos_config.json** file. To do this, insert a new entry in the list of endpoints of type "**strabon**", as highlighted in the following example:

```
{
  "QUERY_RESOLVER_URL": "http://obeos.spaceapplications.com/qa-
resto/rest/2.1/api/query/analyze.json",
  "ONTOLOGY_SERVICE_URL": "http://obeos.spaceapplications.com/ontologies/",
  "ENDPOINTS": [{
    "type": "strabon",
    "endpoints": [{
      "name": "strabon4",
      "url": "http://obeos.spaceapplications.com/endpoint4/Describe"
    },
    {
      "name": "strabon6",
      "url": "http://obeos.spaceapplications.com/endpoint6/Describe",
      "timeout" : 60
    }]
  }, {
    "type": "fedeo",
    "endpoints": []
  }]
}
```

The **timeout** value indicates the time in seconds after which a pending request will be cancelled.

3.7.3 Registration of the FedEO Endpoint

The endpoint of the OBEOS instance of the FedEO OpenSearch Gateway may be configured by editing the **obeos_config.json** file. To do this, edit the URL of the only "**fedeo**" endpoint entry, as highlighted in the following example:

```
{
  "QUERY_RESOLVER_URL": "http://obeos.spaceapplications.com/qa-
resto/rest/2.1/api/query/analyze.json",
  "ONTOLOGY_SERVICE_URL": "http://obeos.spaceapplications.com/ontologies/",
  "ENDPOINTS": [{
    "type": "strabon",
    "endpoints": [{
      "name": "strabon4",
      "url": "http://obeos.spaceapplications.com/endpoint4/Describe"
      "timeout": 60
    }]
  }, {
    "type": "fedeo",
    "endpoints": [{
      "name": "fedeo",
      "dcat": "http://obeos.esa.int/opensearch/"
      "timeout" : 60
    }]
  }]
}
```

The **timeout** value indicates the time in seconds after which a pending request will be cancelled.