User Guide

Following is a stepwise manual to install necessary files and make the software running.

1. Download and install virtuoso opensource 7 for setting up local endpoint where necessary files will be loaded and queried via SPARQL.
For building a local Virtuoso Opensource 7 refer http://virtuoso.openlinksw.com/dataspace/doc/dav/wiki/Main/VOSDownload.

For Virtuoso installation follow following steps:

1. git clone http://github.com/openlink/virtuoso-opensource.git
2. ./autogen.sh
3. ./configure
4. make
5. make install
6. cd var/lib/virtuoso/db
7. virtuoso-t -f &

There is a major difference between virtuoso opensource 7 and 6.1. 6.1 can be installed from the package manager but Virtuoso 7 has to be built on your system.
So installing virtuoso can be a little bit of pain.

2. Once installed, goto http://localhost:8890/conductor/ from your browser. Then log into the database with user id: dba and password: dba. The virtuoso can also be checked, just go to http://localhost:8890/sparql then do a sparql query for the graphs in your endpoint:
SELECT DISTINCT ?graph WHERE {GRAPH ?graph {?s ?p ?o}}. It should return a graphs existing in database.

3. Now change the following values in /var/lib/virtuoso/db/virtuoso.ini, the performance tuning stuff is according to http://virtuoso.openlinksw.com/dataspace/dav/wiki/Main/VirtRDFPerformanceTuning:

```
# note: virtuoso ignores lines starting with whitespace and stuff after a ;
[Parameters]
# you need to include the directory where your datasets will be downloaded
# to, in our case /home/akswadmin/dbpedia_files:
DirsAllowed = ., /usr/share/virtuoso/vad, /usr/local/data/datasets
# IMPORTANT: for performance also do this
[Parameters]
# the following two are as suggested by comments in the original .ini
# file in order to use the RAM on your server:
NumberOfBuffers = 2720000
MaxDirtyBuffers = 2000000
# each buffer caches a 8K page of data and occupies approx. 8700 bytes of
# memory. It's suggested to set this value to 65 % of ram for a db only server
# so if you have 32 GB of ram: 32*1000^3*0.65/8700 = 2390804
# default is 2000 which will use 16 MB ram ;)
# Make sure to remove whitespace if you uncomment existing lines!
```

```
[Database]
MaxCheckpointRemap = 625000
# set this to 1/4th of NumberOfBuffers
[SPARQL]
ShortenLongURIs = 1
```

4. Now restart VOS.

3. Keep the VOS running and from other terminal access the VOS using isql-v or isql-t or isql-vt.
To access VOS from isql :

```
isql-vt 1111 dba dba
```

```
#create Graph
sparql create graph <http://dbpedia.org/> ;
```

#load data into graph - the file must be in an allowed dir defined by "DirsAllowed" in the /var/lib/virtuoso/db/virtuoso.ini

```
#to delete a graph
sparql CLEAR GRAPH <http://dbpedia.org/> ;
sparql drop graph <http://dbpedia.org/> ;
```

5. Now everything is set and we can proceed to run the software. One needs to install all the python packages required in the Software documentation to run the scripts. Both python 2.7 and python 3.3 are required for running the scripts.

6. Now run the bash script to and all the process shall be done automatically for the languages you have specified in the script.

How bash scripts works ?

The bash script is mainly divided into three part:

- run the downloader script
- log into virtuoso, create a graph and load all the necessary files into virtuoso
- run the extractor script
- log into virtuoso and remove the graph created
- remove the files downloaded by the downloader script

<u>Downloader Script</u>

The downloader script downloads the necessary files (given in software documentation) from the DBpedia data dump. These files are in bzip format so needs to be unzipped. The files are stored in a folder named as the language and the decompressed files are in data folder contained in the language folder.
The script decompresses only those abbreviations which end with a ‘.’, ‘!’ or ‘?’ .

<u>Loading Data into Virtuoso</u>

Log into virtuoso database (see to it that VOS is running in the background) from isql as

- isql-v 1111 dba dba
- create a graph - sparql create graph <http://dbpedia.org/> ;
- load all the files - ld_dir_all('address of the unzipped files'', '*.*', 'http://dbpedia.org');
- to check the current status of files to be loaded - select * from DB.DBA.LOAD_LIST;
- run the RDF loader - rdf_loader_run();

Extractor Script

The extractor script reads the abbreviations from the redirects files of respective languages.
The redirects file contains abbreviations URI as triples.
Eg: <http://dbpedia.org/resource/A.D.> <http://dbpedia.org/ontology/wikiPageRedirects> <http://dbpedia.org/resource/Anno_Domini> .
These come in two flavours. First one redirects the resource, i.e. "A.D.", to a page that contains the meaning of the resource, i.e. "Anno_Domini". The second one, for example in this triple:
<http://dbpedia.org/resource/A.I.> <http://dbpedia.org/ontology/wikiPageRedirects> <http://dbpedia.org/resource/Ai> .
links the abbreviation resource to a page that contains a number of disambiguation links, because the abbreviation has more than one meaning.
These informations are retrieved from the local endpoint via SPARQL queries and are stored in different lists which are further used to make lemon and tsv files. The data corresponding to different abbreviations are stored in a dictionary so that it can be referenced with ease.
The queries have been described in the script. At the end all the data is written on a file in lemon format and also in tsv format.
The data output of TSV is in this format:

```
Abbreviation	Definition	Label	Reference Link	sameAs	Type	Category
```

The lemon file format is as follows:

```
@prefix :  <http://nlp.dbpedia.org/abbrevbase> .
@prefix lemon: <http://lemon-model.net/lemon#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix dcterms: <http://purl.org/dc/terms/> .


<http://nlp.dbpedia.org/abbrevbase/lexicon/en>
        lemon:language "en" ;
        lemon:entry : <http://nlp.dbpedia.org/abbrevbase/lexicon/en/entry/A.D.>, <http://nlp.dbpedia.org/abbrevbase/lexicon/en/entry/A.I.> .
# similarly entries for all the abbreviations

<http://nlp.dbpedia.org/abbrevbase/lexicon/en/entry/A.D.>
  lemon:form <http://nlp.dbpedia.org/abbrevbase/lexicon/en/entry/A.D.#form> ;
  lemon:sense <http://nlp.dbpedia.org/abbrevbase/lexicon/en/entry/A.D.#sense> ;
        a lemon:LexicalEntry .

<http://nlp.dbpedia.org/abbrevbase/lexicon/en/entry/A.D.#form>
  lemon:writtenRep "A.D."@en ;
  a lemon:LexicalForm .
```

```
<http://nlp.dbpedia.org/abbrevbase/lexicon/en/entry/A.D.#sense>
   lemon:definition [
            lemon:value "Anno Domini"@en
   ] ;
   rdf:type <http://dbpedia.org/ontology/Place>,<http://dbpedia.org/ontology/PopulatedPlace>,<http:/
/dbpedia.org/ontology/Settlement>,<http://schema.org/Place>  ;
   rdfs:label "Anno Domini" ;
   dcterms:subject <http://dbpedia.org/resource/Category:6th-century_Christianity>,<http:/
/dbpedia.org/resource/Category:Religion_timelines>,<http://dbpedia.org/resource/
Category:Christianity-related_controversies>,<http://dbpedia.org/resource/
Category:Chronology>,<http://dbpedia.org/resource/Category:Christian_terms>,<http://dbpedia.org/
resource/Category:Time>,<http://dbpedia.org/resource/Category:Calendar_eras>,<http://dbpedia.org/
resource/Category:Latin_religious_phrases>  ;
   owl:sameAs <http://el.dbpedia.org/resource/Μ.Χ.>,<http://it.dbpedia.org/resource/
Anno_Domini>,<http://pt.dbpedia.org/resource/Anno_Domini>,<http://cs.dbpedia.org/resource/
Anno_Domini>,<http://de.dbpedia.org/resource/Anno_Domini>,<http://fr.dbpedia.org/resource/
Anno_Domini>,<http://ru.dbpedia.org/resource/От_Рождества_Христова>,<http://es.dbpedia.org/
resource/Anno_Domini>,<http://ja.dbpedia.org/resource/西暦>,<http://pl.dbpedia.org/resource/
Naszej_ery>,<http://ko.dbpedia.org/resource/서력기원>,<http://nl.dbpedia.org/resource/
Anno_Domini>

<http://nlp.dbpedia.org/abbrevbase/lexicon/en/entry/A.I.>
   lemon:form <http://nlp.dbpedia.org/abbrevbase/lexicon/en/entry/A.I.#form> ;
   lemon:sense <http://nlp.dbpedia.org/abbrevbase/lexicon/en/entry/A.I.#sense1>, <http://
nlp.dbpedia.org/abbrevbase/lexicon/en/entry/A.I.#sense2>, <http://nlp.dbpedia.org/abbrevbase/
lexicon/en/entry/A.I.#sense3>, <http://nlp.dbpedia.org/abbrevbase/lexicon/en/entry/A.I.#sense4>,
<http://nlp.dbpedia.org/abbrevbase/lexicon/en/entry/A.I.#sense5>, <http://nlp.dbpedia.org/
abbrevbase/lexicon/en/entry/A.I.#sense6>, <http://nlp.dbpedia.org/abbrevbase/lexicon/en/entry/
A.I.#sense7>, <http://nlp.dbpedia.org/abbrevbase/lexicon/en/entry/A.I.#sense8>,
and so on for all the disambiguations
           a lemon:LexicalEntry .

<http://nlp.dbpedia.org/abbrevbase/lexicon/en/entry/A.I.#form>
           lemon:writtenRep "A.I."@en ;
           a lemon:LexicalForm .

<http://nlp.dbpedia.org/abbrevbase/lexicon/en/entry/A.I.#sense1>
           lemon:definition [
                     lemon:value "Ai (Bible)"@en
           ] ;
           rdf:type <http://schema.org/Event>,<http://dbpedia.org/ontology/Event>,<http://dbpedia.org/
ontology/MilitaryConflict> ;
           rdfs:label "Ai (Bible)" ;
           dcterms:subject <http://dbpedia.org/resource/Category:Torah_cities>,<http://dbpedia.org/
resource/Category:Canaanite_cities>,<http://dbpedia.org/resource/
Category:Hebrew_Bible_cities>,<http://dbpedia.org/resource/Category:Book_of_Joshua> ;
```

```
	owl:sameAs <http://fr.dbpedia.org/resource/Aï_(Bible)>,<http://pl.dbpedia.org/resource/Aj_(miasto_biblijne)>,<http://pt.dbpedia.org/resource/Ai_(Bíblia)>, and so on ;
	a lemon:LexicalSense .

<http://nlp.dbpedia.org/abbrevbase/lexicon/en/entry/A.I.#sense2>
	lemon:definition [
		lemon:value "Ai (poet)"@en
	] ;
	rdf:type <http://schema.org/Person>,<http://dbpedia.org/ontology/Artist>,<http://dbpedia.org/ontology/Writer>,<http://dbpedia.org/ontology/Agent>,<http://dbpedia.org/ontology/Person> ;
	rdfs:label "Ai (poet)" ;
	dcterms:subject <http://dbpedia.org/resource/Category:American_people_of_Irish_descent>,<http://dbpedia.org/resource/Category:English-language_poets>,<http://dbpedia.org/resource/Category:National_Book_Award_winners>,<http://dbpedia.org/resource/Category:University_of_Arizona_alumni>,and so on ;
	owl:sameAs <http://pl.dbpedia.org/resource/Ai_(poetka)>,<http://de.dbpedia.org/resource/Ai_(Dichterin)>,<http://fi.dbpedia.org/resource/Ai_(runoilija)> ;
	lemon:reference <http://dbpedia.org/resource/Ai_(poet)> ;
	a lemon:LexicalSense .
```

and so on for all the sense and abbreviations.

Remove the graph and the files downloaded

The script then deletes the graph and all the files downloaded.

The special characters are handled by UTF-8 encoding and decoding.

References

For detailed information refer the following links

- Lemon tutorial - http://lemon-model.net/
- Loading Bulk Data into VOS - http://joernhees.de/blog/2014/04/23/setting-up-a-local-dbpedia-3-9-mirror-with-virtuoso-7/