

Store CDISC RDF as RRDF data model

PhuseSubTeamAnalysisResults@example.org

2016-06-29

Contents

Store CDISC RDF as RRDF data model	1
Preliminaries	1
Directory for storing rdf.cdisc.org and RDF data cube specification	1
Make local copy of the rdf.cdisc.org	1
Finding the files	2
R-code	2
Setup	2
Create local version using RRDF from local directory containing rdf.cdisc.org	2

Store CDISC RDF as RRDF data model

Preliminaries

The path to the package root is ../../ relative to the present working directory.

```
library(rrdfancillary)
devtools::load_all(pkg="../../")
```

```
## Loading rrdfcdisc
```

Directory for storing rdf.cdisc.org and RDF data cube specification

```
echo change below to suit your setup
export CDISCRDF_HOME=~/projects/phrmwg
```

Make local copy of the rdf.cdisc.org

Here is how to make a local copy of the rdf.cdisc.org - with Bash Shell on Linux. In my setup the ~/projects/phrmwg directory contains a git pull of [rdf.cdisc.org].

```
echo only if directory does not exist
mkdir -p ${CDISCRDF_HOME}
cd ${CDISCRDF_HOME}

git clone https://github.com/phuse-org/rdf.cdisc.org.git
```

Finding the files

Replace directory for find with the appropriate directory for your setup. Also change below for the function `Get.filenames.for.cdsc.standards.R`.

This is one way to find the file names:

```
find ${CDISCRDF_HOME} -name "*.rdf" -o -name "*.owl" -o -name "*.ttl"
```

The result of the function call `Get.filenames.for.cdsc.standards()` in the `rrdfqbcrdn0` package is intended to match the output from `find`.

```
Get.filenames.for.cdsc.standards()
```

Note: this is not a straightforward way to do it.

R-code

Setup

```
library(rrdf)
library(tools)
```

Create local version using RRDF from local directory containing `rdf.cdsc.org`

To conserve space in the R package the files are located outside the package.

The files are stored beneath the directory given by `fileStemDirectory`. The path relative to `fileStemDirectory` is stored in `filesToLoad`.

```
fileStemDirectory<- "~/projects/phrmwg"

filesToLoad<- c(
  "rdf.cdsc.org/resources/w3.org/skos.rdf",
  "rdf.cdsc.org/resources/dublincore.org/dcam.rdf",
  "rdf.cdsc.org/resources/dublincore.org/dcelements.rdf",
  "rdf.cdsc.org/resources/dublincore.org/dcterms.rdf",
  "rdf.cdsc.org/terminology-2013-06-28/glossary-terminology.owl",
  "rdf.cdsc.org/terminology-2013-06-28/cdash-terminology.owl",
  "rdf.cdsc.org/terminology-2013-06-28/sdtm-terminology.owl",
  "rdf.cdsc.org/terminology-2013-06-28/qs-terminology.owl",
  "rdf.cdsc.org/terminology-2013-06-28/send-terminology.owl",
  "rdf.cdsc.org/terminology-2013-06-28/adam-terminology.owl",
  "rdf.cdsc.org/std/sdtm-1-2.ttl",
  "rdf.cdsc.org/std/all-standards.ttl",
  "rdf.cdsc.org/std/cdash-1-1.ttl",
  "rdf.cdsc.org/std/sdtmig-3-1-3.ttl",
  "rdf.cdsc.org/std/sdtmig-3-1-2.ttl",
  "rdf.cdsc.org/std/adamig-1-0.ttl",
  "rdf.cdsc.org/std/sendig-3-0.ttl",
```

```
"rdf.cdisc.org/std/sdtm-1-3.ttl",
"rdf.cdisc.org/std/adam-2-1.ttl",
"rdf.cdisc.org/schemas/ct-schema.owl",
"rdf.cdisc.org/schemas/meta-model-schema.owl",
"rdf.cdisc.org/schemas/cdisc-schema.owl"
)
```

Then the files are input and the size of the resulting file is reported.

```
cdisc.save.zip<- Create.cdisc.standards.from.local(
  cdisc.files.dir= fileStemDirectory,
  CDISCfilelist= filesToLoad
)
```

```
## Loading /home/ma/projects/phrmwg/rdf.cdisc.org/resources/w3.org/skos.rdf ..

## .. total number of triples: 255

## Loading /home/ma/projects/phrmwg/rdf.cdisc.org/resources/dublincore.org/dcam.rdf ..

## .. total number of triples: 275

## Loading /home/ma/projects/phrmwg/rdf.cdisc.org/resources/dublincore.org/dcelements.rdf ..

## .. total number of triples: 417

## Loading /home/ma/projects/phrmwg/rdf.cdisc.org/resources/dublincore.org/dcterms.rdf ..

## .. total number of triples: 1278

## Loading /home/ma/projects/phrmwg/rdf.cdisc.org/terminology-2013-06-28/glossary-terminology.owl ..

## .. total number of triples: 2281

## Loading /home/ma/projects/phrmwg/rdf.cdisc.org/terminology-2013-06-28/cdash-terminology.owl ..

## .. total number of triples: 3209

## Loading /home/ma/projects/phrmwg/rdf.cdisc.org/terminology-2013-06-28/sdtm-terminology.owl ..

## .. total number of triples: 52956

## Loading /home/ma/projects/phrmwg/rdf.cdisc.org/terminology-2013-06-28/qs-terminology.owl ..

## .. total number of triples: 78911

## Loading /home/ma/projects/phrmwg/rdf.cdisc.org/terminology-2013-06-28/send-terminology.owl ..
```

```
## .. total number of triples: 117362

## Loading /home/ma/projects/phrmwg/rdf.cdisc.org/terminology-2013-06-28/adam-terminology.owl ..

## .. total number of triples: 117545

## Loading /home/ma/projects/phrmwg/rdf.cdisc.org/std/sdtm-1-2.ttl ..

## .. total number of triples: 119021

## Loading /home/ma/projects/phrmwg/rdf.cdisc.org/std/all-standards.ttl ..

## .. total number of triples: 119031

## Loading /home/ma/projects/phrmwg/rdf.cdisc.org/std/cdash-1-1.ttl ..

## .. total number of triples: 126475

## Loading /home/ma/projects/phrmwg/rdf.cdisc.org/std/sdtmig-3-1-3.ttl ..

## .. total number of triples: 138761

## Loading /home/ma/projects/phrmwg/rdf.cdisc.org/std/sdtmig-3-1-2.ttl ..

## .. total number of triples: 149523

## Loading /home/ma/projects/phrmwg/rdf.cdisc.org/std/adamig-1-0.ttl ..

## .. total number of triples: 154288

## Loading /home/ma/projects/phrmwg/rdf.cdisc.org/std/sendig-3-0.ttl ..

## .. total number of triples: 162458

## Loading /home/ma/projects/phrmwg/rdf.cdisc.org/std/sdtm-1-3.ttl ..

## .. total number of triples: 164236

## Loading /home/ma/projects/phrmwg/rdf.cdisc.org/std/adam-2-1.ttl ..

## .. total number of triples: 164264

## Loading /home/ma/projects/phrmwg/rdf.cdisc.org/schemas/ct-schema.owl ..

## .. total number of triples: 164340

## Loading /home/ma/projects/phrmwg/rdf.cdisc.org/schemas/meta-model-schema.owl ..
```

```
## .. total number of triples: 164524

## Loading /home/ma/projects/phrmwg/rdf.cdisc.org/schemas/cdisc-schema.owl ..

## .. total number of triples: 164711

## Final rdf.cdisc.org rrdf store, number of triples: 164711

## rrdf store saved to turle file: /tmp/Rtmp3eHW/cdisc-rdf.ttl

## Warning in normalizePath(full.cdisc.save.zip): path[1]="/home/ma/projects/
## rrdfqbcrnd0/rrdfcdisc/inst/extdata/CDISC-standards-rdf/cdisc-rdf.zip": No
## such file or directory

## Writing to zip file /home/ma/projects/rrdfqbcrnd0/rrdfcdisc/inst/extdata/CDISC-standards-rdf/cdisc-r
cdisc.save.zip.info<- file.info(cdisc.save.zip)
message("File ", normalizePath(cdisc.save.zip), " created ", cdisc.save.zip.info$ctime, " size ", cdisc

## File /home/ma/projects/rrdfqbcrnd0/rrdfcdisc/inst/extdata/CDISC-standards-rdf/cdisc-rdf.zip created
```

Creating SPARQL construct script for getting CDISC standard using FROM dataset

The following is experimental and not needed for the present package. See below for explanation.

```
rdf.cdisc.org.URLstem<- "https://github.com/phuse-org/rdf.cdisc.org/raw/master"
rdf.data.cube.URLstem<- "http://publishing-statistical-data.googlecode.com/svn/trunk/specs/src/main/voc
rqfilesToLoad<- filesToLoad
rqfilesToLoad<- gsub("rdf.cdisc.org", rdf.cdisc.org.URLstem, rqfilesToLoad )
rqfilesToLoad<- gsub("rdf-data-cube", rdf.data.cube.URLstem, rqfilesToLoad )

SPARQLscript<- paste(
  "CONSTRUCT { ?s ?p ?o }",
  paste0( "FROM ", "<", rqfilesToLoad, ">", collapse="\n" )
  ,
  "WHERE { ?s ?p ?o }", sep="\n", collapse="\n"
)

cat(SPARQLscript,"\n")
```

```
CONSTRUCT { ?s ?p ?o } FROM https://github.com/phuse-org/rdf.cdisc.org/raw/master/resources/w3.org/skos.rdf FROM https://github.com/phuse-org/rdf.cdisc.org/raw/master/resources/dublincore.org/dcam.rdf
FROM https://github.com/phuse-org/rdf.cdisc.org/raw/master/resources/dublincore.org/dcelements.rdf
FROM https://github.com/phuse-org/rdf.cdisc.org/raw/master/resources/dublincore.org/dcterms.rdf FROM
https://github.com/phuse-org/rdf.cdisc.org/raw/master/terminology-2013-06-28/glossary-terminology.owl
FROM https://github.com/phuse-org/rdf.cdisc.org/raw/master/terminology-2013-06-28/cdash-terminology.
owl FROM https://github.com/phuse-org/rdf.cdisc.org/raw/master/terminology-2013-06-28/sdtm-terminology.
owl FROM https://github.com/phuse-org/rdf.cdisc.org/raw/master/terminology-2013-06-28/qs-terminology.
owl FROM https://github.com/phuse-org/rdf.cdisc.org/raw/master/terminology-2013-06-28/send-terminology.
```

```
owl FROM https://github.com/phuse-org/rdf.cdisc.org/raw/master/terminology-2013-06-28/adam-terminology.
owl FROM https://github.com/phuse-org/rdf.cdisc.org/raw/master/std/sdtm-1-2.ttl FROM https://github.
com/phuse-org/rdf.cdisc.org/raw/master/std/all-standards.ttl FROM https://github.com/phuse-org/rdf.
cdisc.org/raw/master/std/cdash-1-1.ttl FROM https://github.com/phuse-org/rdf.cdisc.org/raw/master/
std/sdtmig-3-1-3.ttl FROM https://github.com/phuse-org/rdf.cdisc.org/raw/master/std/sdtmig-3-1-2.ttl
FROM https://github.com/phuse-org/rdf.cdisc.org/raw/master/std/adamig-1-0.ttl FROM https://
github.com/phuse-org/rdf.cdisc.org/raw/master/std/sendig-3-0.ttl FROM https://github.com/phuse-org/
rdf.cdisc.org/raw/master/std/sdtm-1-3.ttl FROM https://github.com/phuse-org/rdf.cdisc.org/raw/master/
std/adam-2-1.ttl FROM https://github.com/phuse-org/rdf.cdisc.org/raw/master/schemas/ct-schema.owl
FROM https://github.com/phuse-org/rdf.cdisc.org/raw/master/schemas/meta-model-schema.owl FROM
https://github.com/phuse-org/rdf.cdisc.org/raw/master/schemas/cdisc-schema.owl WHERE { ?s ?p ?o }
```

```
# change to be part of package
SPARQLscriptfn<- file.path(tempdir(),"get-rdf-disc-org.rq")
writeLines( SPARQLscript, con=SPARQLscriptfn )
cat("SPARQL script stored in ", normalizePath(SPARQLscriptfn), "\n")
```

SPARQL script stored in /tmp/Rtmp3eHW/get-rdf-disc-org.rq

```
targetDir<- system.file("extdata/CDISC-standards-rdf", package="rrdfcdisc")
targetFile<- file.path(targetDir,basename(SPARQLscriptfn) )

if (file.copy( SPARQLscriptfn, targetFile, overwrite=TRUE)) {
  cat("SPARQL script copied to ", normalizePath(targetFile), "\n")
}
```

SPARQL script copied to /home/ma/projects/rrdfqbernd0/rrdfcdisc/inst/extdata/CDISC-standards-rdf/get-rdf-disc-org.rq

ToDo(mja): The same approach could be used for an SPARQL update load script - that could make a load for each file.

Using the script with R

The R-code below does not work with rrdf.

```
cdisc.rdf.store <- new.rdf()
results <- construct.rdf(cdisc.rdf.store, SPARQLscript )
summarize.rdf(results)
```

The code below assumes a locate fuseki instance is running.

```
endpoint<- "http://localhost:3030/arm/query"
results.fuseki <- construct.remote(endpoint, SPARQLscript )
summarize.rdf(results.fuseki)
```

This worked partially with fuseki 2.3 as of 28-oct-2015: 42199 triples was retrieved. The fuseki endpoint was started using the configuration file ../extdata/sample-rdf/fuseki-crnd-example-config.ttl (see instruction in top of the file).

Alternative a virtouso endpoint can be used:

```
endpoint2<- "http://localhost:8890/sparql"
results.virtouso <- construct.remote(endpoint2, SPARQLscript )
summarize.rdf(results.virtouso)
```

This worked partially with virtuoso 7.2 as of 28-oct-2015: 164976 triples was retrieved. The fuseki endpoint was started using the configuration file `../extdata/sample-rdf/fuseki-crnd-example-config.ttl` (see instruction in top of the file). Note: the virtuoso endpoint must support remote retrieval of data (see http://localhost:8890/sparql?help=enable_sponge) and the value of `ResultSetMaxRows` in the SPARQL section of `virtuoso.ini` should be sufficiently high; for the `rd.f.cdisc.org` mentioned here a value of 200000 may be needed.

Using the script with jena arq

The generated SPARQL query works with arq when using a local endpoint. Replace *get-rdf-disc-org.rq* with the path to the generated file (see above).

Execute the commands from a temporary directory (*/tmp* is used below).

The script below will create a turtle file *construct-rdf.disc.org.ttl* in the

```
export JENAROOT=/opt/apache-jena-3.0.0
export PATH=$PATH:$JENAROOT/bin
arq --query=get-rdf-disc-org.rq > construct-rdf-disc-org.ttl
```

The next statements loads the turtle files generated by the arq invocation above.

```
arqCDISCTtl<- file.path("/tmp", "construct-rdf-disc-org.ttl")
cdisc.rdf.arq.0<- new.rdf(ontology=FALSE)
cdisc.rdf.arq<- load.rdf( arqCDISCTtl, format="TURTLE", appendTo=cdisc.rdf.arq.0)
summarize.rdf(cdisc.rdf.arq)
```

This worked as expected as of 08-mar-2016.

However, the sparql query does not work executed from the Apache Jena Fuseki Version 2.0 or Version 2.3.1. This may be due to some configuration option.