

ICD11 Mapping

Overview

Information on the ICD11 mapping project.

- ModuleId: 123456789
- RefsetId: icd11RefsetId

Documentation

- SNOMED Confluence: [Internal Documentation - ICD-11 Map](#) (need permission)
 - Reviews: <https://confluence.ihtsdotools.org/display/IAP/ICD-11+Algorithm+Reviews>
- [ICD11 Regression Set](#) - This defines concepts reviewed, the target codes they should have so that it can be used to validate against changes in the algorithm.
- [Technical Documentation](#)

Deployment Server Details

- PROD server: prod-mapping-icd11.ihtsdotools.org
 - url: <https://prod-mapping-icd11.ihtsdotools.org/#>
 - cron jobs

```
# Run every day
45 0 * * 0,1,2,3,4,5,6 csh /home/ihtsdo/config/bin/dailyReports.csh > /home/ihtsdo/logs
/dailyReports.log 2>&1
```

- UAT server:
 - url
 - cron jobs - n/a

Software installed

- csh, git, maven

Uploading files

- Log in sftp as you
- Upload files to home dir
- Log in ssh as you
- move files to /tmp
- sudo su
- su -s /bin/bash mapping-service
- cd /opt/mapping-data/ICD11
- cp /tmp/*txt .

Making a Release

- Log into prod-mapping-icd11.ihtsdotools.org
- Become the root user:

```
# this will require having sudo access, then enter your password
sudo su
```

- Become the "mapping-service" user

```
su -s /bin/bash mapping-service
```

- Go to the admin directory for running a release

```
cd /opt/mapping-data/code/admin/release
```

- Make a release

```
time=20171201
mvn install -PRelease -D$src -Drefset.id=icd11RefsetId -Doutput.dir=. -Dtime=$time -Dmodule.
id=900000000000207008 -Dtest.mode.flag=true
# the output will be a file like the following
der2_iisssccRefset_ExtendedMapActiveSnapshot_INT_20171107.txt
```

Important details

- Log in as your user (requires key to be configured)
- User should have "sudo" rights
- The server runs as the "mapping-service" user.
 - For a shell as this user

```
sudo su
su -s /bin/bash mapping-service
```

- App details
 - star/stop server "supervisorctl start/stop mapping-service"
 - look at /etc/supervisor/conf.d/mapping-service.conf for details
 - /opt/mapping-service/lib/mapping-rest.war
 - Admin tools in: /opt/mapping-service-admin/admin (run as mapping-service user)
 - /opt/mapping-service-admin/admin/lucene/mvn install -PReindex -Drun.config=/opt/mapping-service/conf/config.properties
 - Data: in /opt/mapping-data
 - conf - configuration files
 - indexes - lucene indexes
 - Logs - /var/log/supervisor
 - STDOUT log: /var/log/tomcat.log
 - mapping-service-stderr---supervisor-79oJH0.log
 - mapping-service-stdout---supervisor-6Jv0gZ.log
 - Instead of tomcat we use webapp-runner.jar which is a tomcat wrapped up. Allows us to package a known tomcat/webapp package.
 - It is installed to /opt/mapping-service with data in /opt/mapping-data and admin in /opt/mapping-admin
 - a Jenkins job exists allowing you to install wrt updating code or what ever - see below
- Jenkins Deploy - <https://jenkins.ihtsdo.tools.org/view/Mapping/job/deploy-to-prod-mapping-icd11/>
 - This just fires off an ansible process
 - Check /var/log/tomcat.log to see what it's doing (while you run jenkins job)

Details to configure:

Data files for this project are in /home/ihtsdo/data/ICD11

- icd11Concepts.txt - faked from the "linearization" file to be oriented around "entityId" while showing the codes.
- icd11ParChd.txt - faked from the "linearization" file to express relationships between codes that have obvious par/chd relationships (as devined by the textual codes but not the entity codes)
- icd11Map.txt - the computed map (from the algorithm)
- icd11MapNotes.txt - the computed map notes (from the algorithm)

These steps were taken to set up the project

- Manually create and configure the project as mapping admin
 - SNOMED to ICD11 Pilot - top level metadata for project
 - Map Leads selected
 - No map specialists
 - Map relations configured
 - Map age ranges configured
 - Scope concepts configured (starter set minus procedures)
- Remake "SNOMED to ICD11" as a copy.
 - Scope definition configured (ICD10 scope, minus starter set, minus procedures)

Helpful Admin Commands

A cycle of redeploying maps involves running the algorithm to generate the "icd11Map.txt" and "icd11MapNotes.txt" files. Assuming everything described above is already configured, the task here is simply to replace the data. This is done with the "reload.csh" script in /home/ihtsdo/data/ICD11/reload.csh.

```
#!/bin/csh -f

set rc = run.config=/opt/mapping-service/conf/config.properties
```

```

if ($#argv != 1) then
    echo "usage: $0 <refsetId>"
    echo "    icd11RefsetId"
    echo "    icd11RefsetId-b"
    exit 1
endif
#set refsetId = icd11RefsetId-b
set refsetId = $1
if ($refsetId != "icd11RefsetId" && $refsetId != "icd11RefsetId-b") then
    echo "ERROR: unexpected value for refsetId: $refsetId"
    exit 1
endif

set adminDir = /opt/mapping-service-admin/admin

echo "-----"
echo "Starting .../bin/date`"
echo "-----"
echo "refsetId: $refsetId"
echo ""

curl "https://prod-mapping-icd11.ihtsdotools.org/" | grep 'Bad Gateway' > /tmp/x.$$
if ($status == 0 && `cat /tmp/x.$$ | wc -l` == 0) then
    echo "STOP the server before running this (as root)"
    echo "    supervisorctl stop mapping-service"
    exit 1
endif

# Stop server
#sudo supervisorctl stop mapping-service

echo " Extract icd11Map.txt into LOW, MED, HIGH"
grep 'FINAL CATEGORY LOW' icd11MapNotes.txt | cut -f 6 | sort -u -o low.txt
grep 'FINAL CATEGORY MEDIUM' icd11MapNotes.txt | cut -f 6 | sort -u -o medium.txt
grep 'FINAL CATEGORY HIGH' icd11MapNotes.txt | cut -f 6 | sort -u -o high.txt
grep 'FINAL CATEGORY NO MAP' icd11MapNotes.txt | cut -f 6 | sort -u >> high.txt

/bin/rm -f icd11Map.{low,medium,high}.txt
perl -ne 'BEGIN {open(I,"low.txt"); while(<I>){chop; $m{$_}=1;} close(I);} @_ =split/\t/; print if $m{$_[5]}; ' icd11Map.txt > icd11Map.low.txt
perl -ne 'BEGIN {open(I,"medium.txt"); while(<I>){chop; $m{$_}=1;} close(I);} @_ =split/\t/; print if $m{$_[5]}; ' icd11Map.txt > icd11Map.medium.txt
perl -ne 'BEGIN {open(I,"high.txt"); while(<I>){chop; $m{$_}=1;} close(I);} @_ =split/\t/; print if $m{$_[5]}; ' icd11Map.txt > icd11Map.high.txt

/bin/rm -f icd11MapNotes.{low,medium,high}.txt
perl -ne 'BEGIN {open(I,"low.txt"); while(<I>){chop; $m{$_}=1;} close(I);} @_ =split/\t/; print if $m{$_[5]}; ' icd11MapNotes.txt > icd11MapNotes.low.txt
perl -ne 'BEGIN {open(I,"medium.txt"); while(<I>){chop; $m{$_}=1;} close(I);} @_ =split/\t/; print if $m{$_[5]}; ' icd11MapNotes.txt > icd11MapNotes.medium.txt
perl -ne 'BEGIN {open(I,"high.txt"); while(<I>){chop; $m{$_}=1;} close(I);} @_ =split/\t/; print if $m{$_[5]}; ' icd11MapNotes.txt > icd11MapNotes.high.txt

echo " Replace notes for maps .../bin/date`"
perl -ne 'BEGIN {open(I,"icd11MapNotes.txt"); while(<I>){@_ =split/\t/; $m{$_[5]}=1;} close(I);} @_ =split/\t/; print unless $m{$_[5]}; ' notes/icd11MapNotes.txt > fix
/bin/mv -f fix notes/icd11MapNotes.txt
cat icd11MapNotes.txt >> notes/icd11MapNotes.txt

# map remover augmented to remove map records for concept ids in a specified input file
echo " Removing old maps .../bin/date`"
cd $adminDir/remover
mvn install -PMapRecords -D$src -DrefsetId=$refsetId -Dinput.file=/opt/mapping-data/ICD11/icd11Map.txt >&!
/tmp/mvn.log
if ($status != 0) then
    cat /tmp/mvn.log | sed 's/^/ /'
    echo "ERROR removing map records"
    exit 1
endif

```

```

echo " Adding new maps and notes ...`/bin/date`"

echo " add MEDIUM records as EDITING_DONE ...`/bin/date`"
cd $adminDir/loader
mvn install -PMapRecords -D$src -Dinput.file=/opt/mapping-data/ICD11/icd11Map.medium.txt \
-Dmember.flag=false -Drecord.flag=true -Dworkflow.status=EDITING_DONE >&! /tmp/mvn.log
if ($status != 0) then
    cat /tmp/mvn.log | sed 's/^/ /'
    echo "ERROR installing maps"
    exit 1
endif
echo " add MEDIUM notes ...`/bin/date`"
cd $adminDir/loader
mvn install -PMapNotes -D$src -Dinput.file=/opt/mapping-data/ICD11/icd11MapNotes.medium.txt >&! /tmp/mvn.log
if ($status != 0) then
    cat /tmp/mvn.log | sed 's/^/ /'
    echo "ERROR installing map notes"
    exit 1
endif

echo " add HIGH as PUBLISHED ...`/bin/date`"
cd $adminDir/loader
mvn install -PMapRecords -D$src -Dinput.file=/opt/mapping-data/ICD11/icd11Map.high.txt \
-Dmember.flag=false -Drecord.flag=true >&! /tmp/mvn.log
if ($status != 0) then
    cat /tmp/mvn.log | sed 's/^/ /'
    echo "ERROR installing maps"
    exit 1
endif
echo " add HIGH notes ...`/bin/date`"
cd $adminDir/loader
mvn install -PMapNotes -D$src -Dinput.file=/opt/mapping-data/ICD11/icd11MapNotes.high.txt >&! /tmp/mvn.log
if ($status != 0) then
    cat /tmp/mvn.log | sed 's/^/ /'
    echo "ERROR installing map notes"
    exit 1
endif

# before doing workflow stuff with HIGH, compute workflow
echo " Compute workflow ...`/bin/date`"
cd $adminDir/loader
mvn install -PComputeWorkflow -D$src -Drefset.id=$refsetId >&! /tmp/mvn.log
if ($status != 0) then
    cat /tmp/mvn.log | sed 's/^/ /'
    echo "ERROR running compute workflow"
    exit 1
endif

echo " convert HIGH to QA ...`/bin/date`"
cd $adminDir/loader
mvn install -PAdHoc -D$src -Drefset.id=$refsetId -Dmode=icd11 -Dinput.file=/opt/mapping-data/ICD11/icd11Map.
high.txt >&! /tmp/mvn.log
if ($status != 0) then
    cat /tmp/mvn.log | sed 's/^/ /'
    echo "ERROR running adhoc step"
    exit 1
endif

# compute workflow again when finished assigning to QA
echo " Compute workflow ...`/bin/date`"
cd $adminDir/loader
mvn install -PComputeWorkflow -D$src -Drefset.id=$refsetId >&! /tmp/mvn.log
if ($status != 0) then
    cat /tmp/mvn.log | sed 's/^/ /'
    echo "ERROR running compute workflow"
    exit 1
endif

```

```
# Start server -  
#sudo supervisorctl start mapping-service  
  
echo "-----"  
echo "Finished ...`/bin/date`"  
echo "-----"  
echo ""  
echo "REMEMBER to restart the server = supervisorctl start mapping-service"
```