

## **Map Record Loader and Remover**

# Overview

This page documents the use of admin tools to load or remove map records. This is used to take mappings from a legacy system and install them into the new environment.

## Prerequisites

- MySQL database must already exist (e.g. "mappingservicedb").
- MySQL database connection parameters must be defined in the properties file specified by "run.config".
- Create/Update DB mojo must also have already run so that the database schema exists.
- A map project must exist with a refset id matching the data intended to be loaded
  - Either create a map project and configure it properly
  - Or import a map project
- If using the "complex map refset" loader, the terminology version with the corresponding complex or extended map records must already be loaded (e.g. SNOMED CT) as well as the destination terminology for that mapping refset (e.g. ICD10).
- **IMPORTANT:** The Tomcat server must not be running while this is run, otherwise indexes can get corrupted.

## Details

These tools are used to load map records from either a loaded terminology (e.g. SNOMED CT) or from an RF2 file. In either case, there is an alignment that must exist between the refset id in the file (or database) and the refset id of one of the map projects defined in the system. Typically this process should only be run once for each map project and if data needs to be reloaded, the remover should be run first.

The complex map loader is used to create initial map record objects based on a terminology that has been loaded. For example, if the 20140731 version of SNOMEDCT is loaded into the tool, this mojo could be used to create map record objects (associated with a particular map project) for one of the complex or extended map refsets defined in that release. All loaded records are marked as "PUBLISHED".

**Mojo:** MapRecordComplexMapLoaderMojo.java (in admin/mojo/src/main/resources/java/org/ihtsdo/otf/mapping/mojo)

**Project:** admin/loader

### Configuration Parameters

- A profile of "CreateMapRecords" must be specified
- run.config - the standard configuration file specified as a -D parameter
- refset.id - a comma-separated list of refset ids to create map records for specified as a -D parameter
  - Map projects with these refset ids must exist.

The RF2 complex map loader is used to load map record objects from an RF2 file into the map tool. Typically, this is used as a transport mechanism for unpublished map records from another system or to reload records after a release. All loaded records are marked as "PUBLISHED" by default.

**Mojo:** MapRecordRf2ComplexMapLoaderMojo.java (in admin/mojo/src/main/resources/java/org/ihtsdo/otf/mapping/mojo)

**Project:** admin/loader

### Configuration Parameters

- A profile of "MapRecords" must be specified
- run.config - the standard configuration file specified as a -D parameter
- record.flag - whether to create MapRecord objects for these complex map entris
  - Set to "true" when loading records for the first time
- member.flag - whether to create ComplexMapRefSetMember objects for these entries
  - Set to "true" when members have been removed (say following a release) and you want to reload them.
  - Generally this flag will be true
- input.file - the input file specified as a -D parameter
  - Map projects with the refset ids specified in this file must exist

## Related

- You may also see the mojo MapRecordRf2ComplexMapSampleLoaderMojo.java. This is a special purpose mojo for loading map records being incrementally developed using other tooling into this tooling environment with a particular sampling ratio. Unless you specifically need this, the mojo should be avoided. It was developed for a special purpose mapping project that required sampling review.

The RF2 simple map loader is used to load map record objects from an RF2 file into the map tool. Typically, this is used as a transport mechanism for unpublished map records from another system or to reload records after a release.. All loaded records are marked as "PUBLISHED" by default.

**Mojo:** MapRecordRf2SimpleMapLoaderMojo.java (in admin/mojo/src/main/resources/java/org/ihtsdo/otf/mapping/mojo)

**Project:** admin/loader

#### Configuration Parameters

- A profile of "SimpleMapRecords" must be specified
- run.config - the standard configuration file specified as a -D parameter
- record.flag - whether to create MapRecord objects for these complex map entreis
  - Set to "true" when loading records for the first time
- member.flag - whether to create ComplexMapRefSetMember objects for these entries
  - Set to "true" when members have been removed (say following a release) and you want to reload them.
  - Generally this flag will be true
- input.file - the input file specified as a -D parameter
  - Map projects with the refset ids specfied in this file must exist

The map record remover is used to remove map records for a map project. This is typically used either in the development environment to reset the state of the tool, or can also be used after a SNOMED CT release to reload the final published view of map records back into the system. The release process bases the delta computation on the internal representation of the complex map refset members so removing and reloading them creates a new baseline for comparsion (for the following release).

**Mojo:** MapRecordRemoverMojo.java (in admin/mojo/src/main/resources/java/org/ihtsdo/otf/mapping/mojo)

**Project:** admin/remover

#### Configuration Parameters

- A profile of "MapRecords" must be specified
- run.config - the standard configuration file specified as a -D parameter
- refset.id - a comma-separated list of refset ids to remove map records for specified as a -D parameter
  - Map projects with these refset ids must exist.

## Samples

Sample command line call of the admin tool to load map records from complex/extended map refset entries for refsets 447563008 and 447562003:

```
% cd ~/code/admin/loader
% mvn install -Drun.config=/home/ihtsdo/data/config.properties -PCreateMapRecords -Drefset.id=447563008,447562003
```

Sample command line call of the admin tool to load map records from an RF2 file:

```
% cd ~/code/admin/loader
% mvn install -Drun.config=~/config/config.properties -PMapRecords -Dinput.file=/home/ihtsdo/data/der2_iisssccRefset_ExtendedMapSnapshot_INT_20140131.txt
```

Sample command line call of the admin tool to remove map records for for refsets 447563008 and 447562003:

```
% cd ~/code/admin/remover
% mvn install -Drun.config=/home/ihtsdo/config/config.properties -PMapRecords -Drefset.id=447563008,447562003
```

Sample Eclipse run configuration to load map records from complex map refset entries for refsets 447563008 and 447562003:

Name: loader - CreateMapRecords 447563008,447562003

Main JRE Refresh Source Environment Common

Base directory:  
 C:/Users/Brian Carlsen/workspace/OTF-Mapping-Service/admin/loader  
 Browse Workspace... Browse File System... Variables...

Goals: clean install Select...

Profiles: CreateMapRecords

Offline  Update Snapshots  
 Debug Output  Skip Tests  Non-recursive  
 Resolve Workspace artifacts

1 Threads

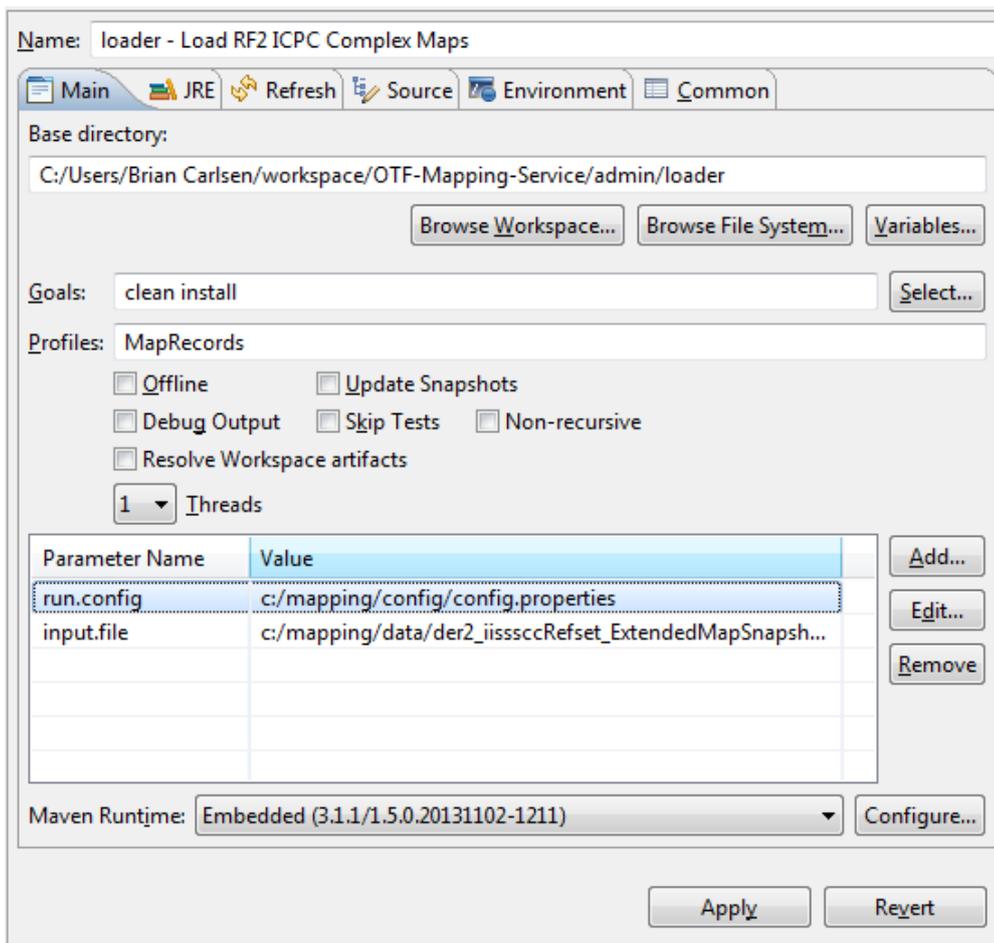
Parameter Name	Value
reset.id	447562003,447563008
run.config	c:/mapping/config/config.prope...

Add... Edit... Remove

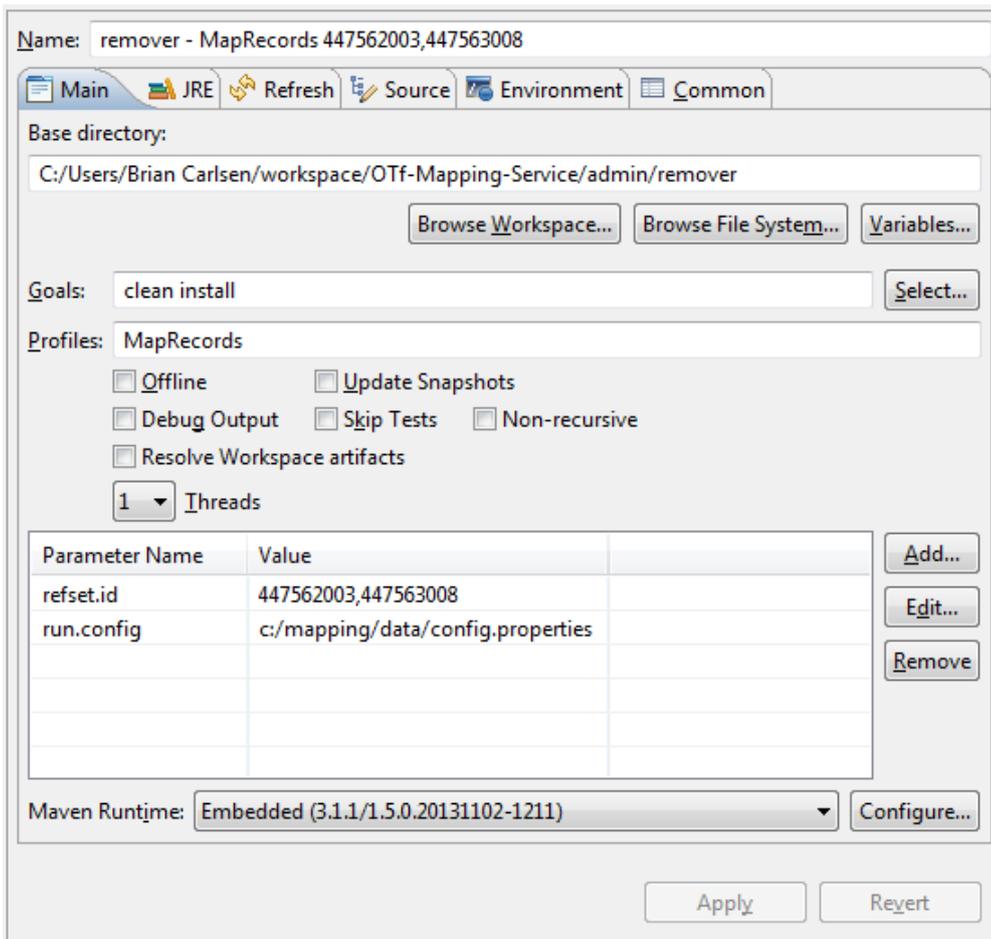
Maven Runtime: Embedded (3.1.1/1.5.0.20131102-1211) Configure...

Apply Revert

Sample Eclipse run configuration to load map records from an RF2 file:



Sample Eclipse run configuration to remove map records for refsets 447563008 and 447562003:



## Troubleshooting

If anything goes wrong while loading map records, the remover can simply be used to clear the state and try again. The main two strategies for loading the mapping tool are

Load a terminology that contains mappings to be further maintained -> in which the "CreateMapRecords" mode is used.

Load records from another system that are not published with the terminology and are now going to be maintained -> in which the approach of loading an RF2 file of the map records is used.

## References/Links

- n/a