

Performing an RF2 Release

Overview

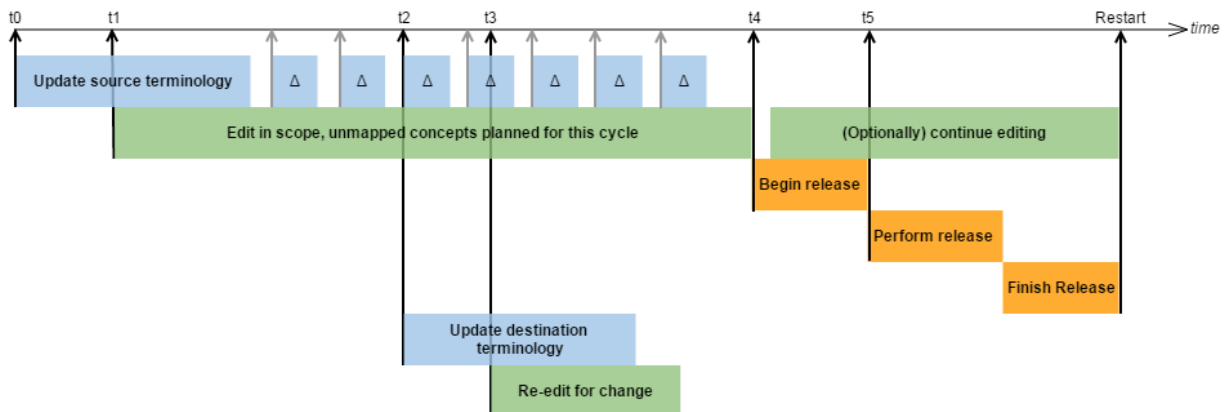
Documents the process of creating an RF2 release.

Prerequisites

- The official release version of the previous RF2 snapshot must be loaded into the `complex_map_refset_members` table. It is against this that a "delta" is computed.
- All editing for the release cycle must be completed (meaning all map records for the project are in "READY_FOR_PUBLICATION" or "PUBLISHED" state).
- All map records are valid.

Details

Release of the mapping data is handled outside of the application via admin tools. The currently implemented process creates a number of RF2 release artifacts in a specified directory. You can see in the editing cycle diagram the recommended process.



The release begins at the planned conclusion of an editing cycle.

Begin Release

This is a way of officially indicating that a release is to start. It does a number of things, including

- Verify that all records for the map project are publication ready (meaning with a workflow status of READY_FOR_PUBLICATION or PUBLISHED).
- Run project-specific validation against all publication ready records
- Generate a "Release QA" report to capture any information about validation failures
- IF there were no validation failures
 - Remove map records for retired and out of scope concepts.
- IF there were failures, go to the application and look at the most recent "Release QA" report for the project
 - NOTE: this report definition must exist for the project for this mechanism to work

If the first steps fail to verify the preconditions, the removing of out of scope map records does not occur. If they are mistakenly removed, information about them can be recovered from the audit tables.

NOTE: this tool can also be run with a `-Dtest.mode.flag=true` parameter that will perform all of the steps except for removing map records - thus allowing the conditions to be tested without making any destructive changes.

Perform/Process Release

This step generates several data artifacts in a specified output directory:

- A delta file relative to the current state of the `complex_map_refset_members` table.
- An active snapshot file showing only the current, active states of the map (with effective time set to the latest version)
- A "human readable" file showing the active snapshot but with additional columns for the source/target terminology names and the map relation name.
- A true snapshot file corresponding to what you would get if you were to combine delta with previous full and generate a new snapshot.
 - This can be used for QA
 - This is the file to load back in during the **Finish Release** step.

- A statistics report for the files needed for documentation.

This process is non-destructive and can be iteratively run as changes are made. In fact, it could also be used to bootstrap a "daily build" that would render a delta relative to the previous release for any state of the database. If any thing goes wrong, the problem can be fixed and the process can simply be re-run.

NOTE: this tool can also be run with a `-Dtest.mode.flag=true` parameter that will avoid full recomputation of concept preferred names, which makes it run a little faster.

Finish Release

This step is run when the release process is complete and no further changes are expected. It performs these operations.

- Mark `READY_FOR_PUBLICATION` records as `PUBLISHED`
- Update the "complex map refset members" table to reflect the published/snapshot state (for future comparisons).

NOTE: this tool can also be run with a `-Dtest.mode.flag=true` parameter that will simulate and report the steps above without actually making any changes. Thus, you can see which records will be marked as `PUBLISHED` before making a decision about when to run this.

Begin Editing Cycle

After a release is finalized and editing is to start again, this tool should be executed to set the "editing cycle begin date" for the map project. Reports and QA checks can perform analysis based on this date to clearly separate edits for the previous release vs. edits for the next release. This mechanism is helpful if editing is to begin prior to the official release of the data.

For more information on running the admin tools to enact a release, see

- [Release Processing Tools](#)

Given that SNOMED updates maps alongside the release, and in fact includes maps in the release, a related page to consider is [Updating a Source Terminology](#).

Daily Release Process

A daily release process could be developed around the "ReleaseProcessMojo". This runs QA and generates release artifacts based on the current state of published and publication ready records. By simply running this without the "begin" or "finish" steps, a daily snapshot could be produced. NOTE: it may be important to disable some of the QA checks if they prevent a release from occurring.

The ideal setup would be something like this:

- Create a "mapping-daily-build" process that would do these things:
 - Run the release processing mojo with an output directory in the target/ folder
 - Use the assembly plugin to package the release files (or maybe just the delta file) into a zip file
 - Attach the .zip file artifact to this module as an artifact
 - Publish the snapshot version of the module to a repository (e.g. <http://nexus.ihtsdo.org>)
- Automate this
 - One option would be to use a CI server configuration
 - Another option would be to create a wrapper script and use a cron job

Continuing to Edit While Running a Release

This is not currently supported, though clever uses of workflow status (and avoiding new "publication ready states" could be easily employed to address this. The IHTSDO editing process aligns maps with the SNOMEDCT release and so this functionality was not needed.

Alternative Release Process

Additional or alternative release processes could easily be implemented that would behave differently. Some things to consider are:

- Not requiring all workflow to be publication ready - simply publishing only things that are ready.
- Validating only things that are publication ready and not trying to validate all records.
- Producing data formats other than RF2 snapshot and RF2 delta.

References/Links

- [Release Processing Tools](#)
- [Updating a Source Terminology](#)