Updating a Source Terminology

Overview

Documents the process for updating a map project source terminology at the point between one editing cycle and the next.

Details

Consider the example of SNOMEDCT. There is a January/July release cycle and maps are updated to be in sync with the release cycles. At the end of one editing cycle, the next version of SNOMED is officially released and the mapping tool should reflect the state of this release. The map projects and terminology content are "version aware", meaning there is a terminologyVersion field that must be maintained. Depending on choices made regarding this value, there are different strategies for ensuring the mapping tool has access to the intended version of the source terminology.

Maintaining terminologyVersion

There are two strategies for using terminology version that are useful for different use cases.

- Use a real terminology version reflecting the actual release.
    - For example, for SNOMEDCT, this would be something like "20140731" or "20150131"
    - This is ideal when the mapping tool is used to maintain mappings from legacy versions of the terminology or when the mapping artifact is itself released on an independent cycle from the underlying terminology.
- Use a meta-value that does not need to be changed from release to release
    - For example, we could use the value "latest"
    - This is ideal when the mapping artifact is itself released as part of the official terminology release (as is the case for SNOMEDCT).
    - This is the default approach used by the RF2 terminology loader, though it can be overridden by using a -Dversion=... flag
    - *This strategy is used for "SNOMEDCT to XXX" mappings.*

Updating the source terminology content and version

There are two strategies here.

- Remove the old version and import the new version
    - This involves using the terminology remover to remove the prior version of the terminology and then completely reloading the release version using the corresponding terminology loader.
    - This is ideal for projects that do not have access to an incremental delta and are mapping terminologies that are released separately from the mapping.
    - This strategy works with either choice for "maintaining terminologyVersion" and guarantees the correctness of the terminology representation.
    - The timing of this is important if you have a map project where editors continue to edit for the next editing cycle before the official terminology release is ready.
- Apply a definitive delta file and (optionally) update the terminologyVersion
    - This strategy only works well if using the "latest" style of "maintaining terminologyVersion".
    - This is ideal for projects that have access to an incremental delta and have mapping data released with the underlying source terminology.
    - *This strategy is used for "SNOMEDCT to XXX" mappings.*

Updating the map project source terminology version

When using an actual terminology version value, when a new version of the terminology is loaded, the map project metadata must be updated to reflect the change in "source terminology version". This can be done by an admin in the map project metadata editing tools.

When using a terminology version value of "latest", it is presumed that the map project metadata is already set to this and does not require updating.

Computing Worfklow

Any time a terminology changes, any map projects using that as a source terminology need to have workflow recomputed. This is done using the compute workflow (see Workflow Tools).

Updating Concept/Target Names

Once everything is updated, use the Concept and Target Name Updater to update the concept name of map records.

Handling Changes in Delta Processing

There is one other tricky wrinkle in all this. If an editing cycle started with a clean load of the previous release version of the terminology and was followed by a series of applications of daily builds (say each Sunday) there's a few data conditions that are not properly handled. In particular, if data elements are changed in one daily build, and then those changes are retracted in a subsequent daily build, there is not enough information to identify and reconstruct the prior state.

Thus, it is recommended that at the end of an editing cycle, the previous release terminology be reloaded from scratch and then the final (definitive) delta applied to that. This will yield exactly the next release and can be used as a basis for computing worfklow one last time and ensuring all in-scope concepts are mapped.

References/Links

- Terminology Loader and Remover