

Identifier Assignment

Overview

Documents identifier management.

Details

There are several kinds of identifiers that are managed in different ways. Each is discussed below.

Hibernate managed identifiers

Each hibernate object has a framework-managed identifier that is a "long" value. These identifiers are completely generated and managed by the framework and so no external maintenance of identifiers is needed. It's an instance-level identifier and not based on any kind of "identity" calculation. Once an object is created, its identity is represented exactly by this id.

Terminology identifiers

All terminology objects have terminologyId, terminology, and terminologyVersion fields. The terminologyId is the release identifier of the object (e.g. a concept id for a Concept, a description id for a Description, and so on). Two domain objects can have the same terminologyId within the same terminology and terminologyVersion if that is supported by the indicated terminology. For RF2 and SNOMED, terminologyId values are unique within a particular version of the terminology. This is also true for ICD9CM, ICD10, and ICPC.

UUIDs for published map records

Map records use UUIDs as their terminology identifiers. Various strategies have been employed over the years to manage these identifiers. This mapping tool takes the approach of a generated UUID based on the "identity" fields of a complex map refset entry. Identity field tuples are unique across a snapshot release of a map project (even considering up propagation). This approach allows movement within group or priority without need to reassign identifiers (thus reducing churn).

Identity fields are defined as:

- refSetId
- referencedComponentId
- mapRule
- mapTarget.

Not all historical map records use this approach, so the release algorithm builds a map of hashed identity fields to prior identifier based on the previous release and reuses identifiers from the past (regardless of how they were generated).

Thus, for ICD9CM, the refSetId, referencedComponentId, mapRule, and mapTarget tuple is guaranteed to be unique among active mappings. An algorithm determines the identifier used by this combination of values and reuses it for the current state. For any new complex map refset entries to be created for the release (e.g. new delta entries for new mappings) the UUID algorithm above is used.

References/Links

- n/a