**U.S. Veterans Administration**

**Chief Health Information Office (CHIO) Blanket Purchase Agreement (BPA) 54 - Collaboration and Tooling**

**Task Order 2 – Workbench Infrastructure**

**August 6, 2013**

**Task 7?: Report documenting the import individual terminologies from the Unified Medical Language System (UMLS) into the International Health Terminology Standards Development Organization's (IHTSDO) Terminology Workbench   
  
Contract: GS35F0009L?**

Table of Contents

[1. Overview 4](#_Toc363597462)

[2. Unified Medical Language System (UMLS) 4](#_Toc363597463)

[2.1. A Brief Overview of the UMLS 4](#_Toc363597464)

[3. Loading UMLS content into the IHTSDO Workbench 4](#_Toc363597465)

[3.1. Obtaining the UMLS content 4](#_Toc363597466)

[3.2. Running MetamorphoSys 5](#_Toc363597467)

[3.2.1. Output Options 5](#_Toc363597468)

[3.2.2. Source List 5](#_Toc363597469)

[3.2.3. Precedence 5](#_Toc363597470)

[3.2.4. Complete Subset 5](#_Toc363597471)

[3.3. Setting up and running the Workbench UMLS Loader 6](#_Toc363597472)

[3.3.1. Getting the transform tools 6](#_Toc363597473)

[3.3.2. Using the transform tools 6](#_Toc363597474)

[3.3.3. Converting the source content 6](#_Toc363597475)

[4. Including the output into a Workbench Bundle 8](#_Toc363597476)

[5. UMLS Terminologies in the Workbench 10](#_Toc363597477)

[5.1. The UMLS Taxonomy in the Workbench 10](#_Toc363597478)

[5.2. UMLS Concepts in the Workbench 11](#_Toc363597479)

[5.2.1. AUIs 11](#_Toc363597480)

[5.2.2. AUI Based Concepts 13](#_Toc363597481)

[5.2.2.1. Arena View 15](#_Toc363597482)

[5.2.3. CUIs 16](#_Toc363597483)

[5.3. Relationships 17](#_Toc363597484)

[5.4. UMLS Metadata 19](#_Toc363597485)

[5.4.1. UMLS Terminology Specific Metadata 19](#_Toc363597486)

[5.4.2. UMLS Shared Metadata 21](#_Toc363597487)

[5.4.3. Refsets 22](#_Toc363597488)

[5.5. Workbench Data Mapping 23](#_Toc363597489)

[6. Appendix A 25](#_Toc363597490)

# Overview

This document describes the process that was created to transform individual terminologies from the UMLS RRF format into a format suitable for the IHTSDO Workbench.

# Unified Medical Language System (UMLS)

## A Brief Overview of the UMLS

The Unified Medical Language System (UMLS) is a compendium of many controlled vocabularies in the biomedical sciences, initially created 1986. It provides a mapping structure among these vocabularies and thus allows one to translate among the various terminology systems; it may also be viewed as a comprehensive thesaurus and ontology of biomedical concepts.

The UMLS was designed and is maintained by the US National Library of Medicine, and is updated quarterly. The project was initiated in 1986 by Donald A. B. Lindberg, M.D., then and current Director of the Library of Medicine.

The current UMLS release (2013AA) includes over 150 classifications, code sets, thesauri and lists of controlled terms from the biomedical domain. The distribution is nearly 4GB while still compressed.

UMLS distribution includes tooling (MetamorphoSys) which is used to extract and subset the UMLS into the Rich Release Format (RRF) [[1]](#footnote-1). The RRF output files are the starting point for loading the UMLS content into the Workbench.

# Loading UMLS content into the IHTSDO Workbench

This chapter covers the process of converting UMLS content into the Workbench format. Users who just want to include already converted UMLS content into their Workbench bundle should skip ahead to Chapter 4.

## Obtaining the UMLS content

With other terminologies which have been converted into the Workbench format for the VA, we have hosted a copy of the native source format within the VA Archiva server.

However, because of the size of the UMLS and the required MetamorphoSys conversion process, we have not hosted a copy of the UMLS in Archiva.

Developers wishing to convert UMLS content will have to download the UMLS and run MetamorphoSys as a prerequisite to converting the content to the Workbench format.

The UMLS content can be downloaded from:

<http://www.nlm.nih.gov/research/umls/licensedcontent/umlsknowledgesources.html>

## Running MetamorphoSys

After downloading the UMLS content and extracting MetamorphoSys (mmsys.zip) the next step is to run the UMLS MetamorphoSys to subset the content. This will create the RRF source files which are required by the Workbench Converter.

The operation of MetamorphoSys will not be covered in depth in this document. Rather, a guide to the required settings for the Workbench converting tooling will be provided. The full documentation for MetamorphoySys tooling can be found on the NLM website:  
<http://www.nlm.nih.gov/research/umls/implementation_resources/metamorphosys/>

After launching MetamorphoSys, select “Install UMLS”. After providing the “Output Destination”, select the “New Configuration” button. A panel with a number of tabs is displayed.

### Output Options

* The ‘Output Format’ must be configured to ‘Rich Release Format’ (which is the default setting)
* The ‘Source Abbreviation Format’ can be set to either ‘versioned’ or ‘unversioned’, depending on which style of Source Attributes are desired in the Workbench.
* The ‘Browser Index Files’ are not required
* The ‘Database Load Scripts’ are not required

### Source List

Any number of sources can be selected for conversion into the Workbench. The Workbench conversion tool also allows users to further subset which source files they want to include in a Workbench Data file when it runs as well.

For the best performance, only export the desired terminologies from MetamorphySys.

The currently published Workbench UMLS data file contains:

* ICD9CM – International Classification of Diseases, Ninth Revision, Clinical Modification
* ICD10CM – International Classification of Diseases, 10th Edition, Clinical Modification
* HCPCS – Healthcare Common Procedure Coding System
* MTHHH – Metathesaurus HCPCS Hierarcial Terms
* CPT – Current Procedural Terminology
* MTHCH – Metathesaurus CPT Hierarchial Terms

### Precedence

The Workbench conversion tool will follow the Precedence specified here when selecting the term description type to use for the FSN and Preferred Synonym.

### Complete Subset

Once configured, launch the Subset process within MetamorphoSys. This may take a significant amount of time to run, depending on your subset choices and your hardware.

## Setting up and running the Workbench UMLS Loader

### Getting the transform tools

The Workbench conversion tooling developed to load the RRF output of MetamorphoSys is hosted here:

<https://csfe.aceworkspace.net/svn/repos/va-oia-terminology-converters/UMLS/trunk/>

The UMLS conversion tooling also depends on a library which is shared with the RxNorm Workbench Data loader (which also processes RRF formatted data files). This shared library is hosted here:

<https://csfe.aceworkspace.net/svn/repos/va-oia-terminology-converters/umlsLoaderUtilities/trunk>

### Using the transform tools

This section documents the process of converting the RRF output from MetamorphySys into the Workbench format. This document assumes that you have a working Maven environment on your system. Installation and configuration of the Maven environment for use with the Workbench is not discussed in this document.

### Converting the source content

1. Build and install the umlsLoaderUtilities project into your local maven environment. No special configuration is required for this project.
2. Configure the converter by editing the file UMLS\UMLS-econcept\pom.xml. The following components of the buildUMLS ‘execution’ can be customized:
   1. <srcDataPath> - Required – This needs to point to the path where MetamorphoSys was configured to place the output. This path should end with a folder that corresponds to the UMLS version number. For example:  
      C:\temp\UMLS\output\2013AA\
   2. <tmpDBPath> - Optional – This variable may be used to specify the path where temporary data will be written during the conversion. This is provided because the temporary data can be quite large, and it may be necessary to place it on a different disk drive, depending on your hardware.
   3. <sabFilters> - Optional – This variable may be used to further subset the content that is converted into the Workbench format. If this variable is omitted, all terminologies that were output by MetamorphoSys will be converted. When this parameter is provided, only terminologies that match the filter will be converted.
      1. Note – the SAB identifiers to use here must match the selection made during the MetamorphoSys process – either ‘versioned’ or ‘unversioned’ SABs.
      2. Note – The SAB ‘SRC’ should always be included (if a sabFilter is provided) so that Hierarchies can be constructed properly during the conversion.
   4. <additionalRootConcepts> - Optional – This parameter can be used to specify additional root concepts that should be treated as the roots of a Hierarchy within the Workbench.
3. Additionally, while editing the file UMLS\UMLS-econcept\pom.xml the appropriate value for <version> should be specified to match the version of the UMLS that was extracted.
4. Execute the command:  
   mvn clean install from within the UMLS folder.  
   This command will take a significant amount of time to run, depending on the number of terminologies included in the conversion. For very large conversions, you may need to increase the Java Heap memory settings for your maven installation.
5. When the conversion process is finished, the resulting file RRF-MR.jbin will be located in the folder UMLS\UMLS-econcept\target\. This folder contains both a jbin file, and a zip file (which also contains the jbin file).
6. After testing the output file locally in a build of the Workbench, the UMLS\UMLS-econcept project can be published to the VA archiva server by executing a mvn deploy command. This will upload the zip file containing the RRF-MR.jbin file and some conversion metadata and statistics. The specific deploy command required will depend on your Maven configuration.

# Including the output into a Workbench Bundle

The converted UMLS content can be included in the Workbench by building it into the Workbench database. Your database project (baseline) pom file will need the following section to declare the dependency and extract the zip file:

<plugin>

<groupId>org.apache.maven.plugins</groupId>

<artifactId>maven-dependency-plugin</artifactId>

<executions>

<execution>

<id>fetch-UMLS-data</id>

<phase>generate-sources</phase>

<goals>

<goal>unpack</goal>

</goals>

<configuration>

<artifactItems>

<artifactItem>

<groupId>gov.va.oia.terminology.converters</groupId>

<artifactId>UMLS-econcept</artifactId>

<version>2013AA-loader-1.0</version>

<type>zip</type>

</artifactItem>

</artifactItems>

<outputDirectory>${project.build.directory}/db</outputDirectory>

</configuration>

</execution>

</executions>

</plugin>

Then, use the load-econcepts-multi execution, and add in the RRF-MR.jbin file to the conceptsFileNames section.

<plugin>

<groupId>org.ihtsdo</groupId>

<artifactId>wb-bdb-mojo</artifactId>

<executions>

<execution>

<id>load-econcepts</id>

<phase>process-sources</phase>

<goals>

<goal>load-econcepts-multi</goal>

</goals>

<configuration>

...

<conceptsFileNames>

<String>eConcepts.jbin</String> <!-- snomed -->

<String>RRF-MR.jbin</String>

</conceptsFileNames>

...

Finally, because a new path is created for the UMLS content, you will want to configure the workbench to include the path as an origin for the current path. The FSN of the path concept is ‘UMLS Path’. One way to configure this path during workbench assembly is to modify the file:

[workbenchAssemblyProject]\database\project-bdb\pom.xml

You will be adding the SimpleUniversalAcePosition section, as documented below.

<profiles>

…

<profile> (init-db)

…

<build>

…

<plugins>

…

<plugin> (wb-mojo)

…

<executions>

…

<execution> (create-baseline-development-path)

…

<configuration>

…

<origins>

…

<SimpleUniversalAcePosition>

<pathConcept>

<pathFsDesc>UMLS Path</pathFsDesc>

<description>UMLS Path</description>

</pathConcept>

<timeStr>latest</timeStr>

</SimpleUniversalAcePosition>

</origins>

# UMLS Terminologies in the Workbench

## The UMLS Taxonomy in the Workbench

In order to browse UMLS terminologies in the workbench, you may need to add the root node “UMLS Root Concepts” into the workbench taxonomy configuration. Search for “UMLS Root Concepts” and drag the result into the taxonomy configuration dialog, as shown below.

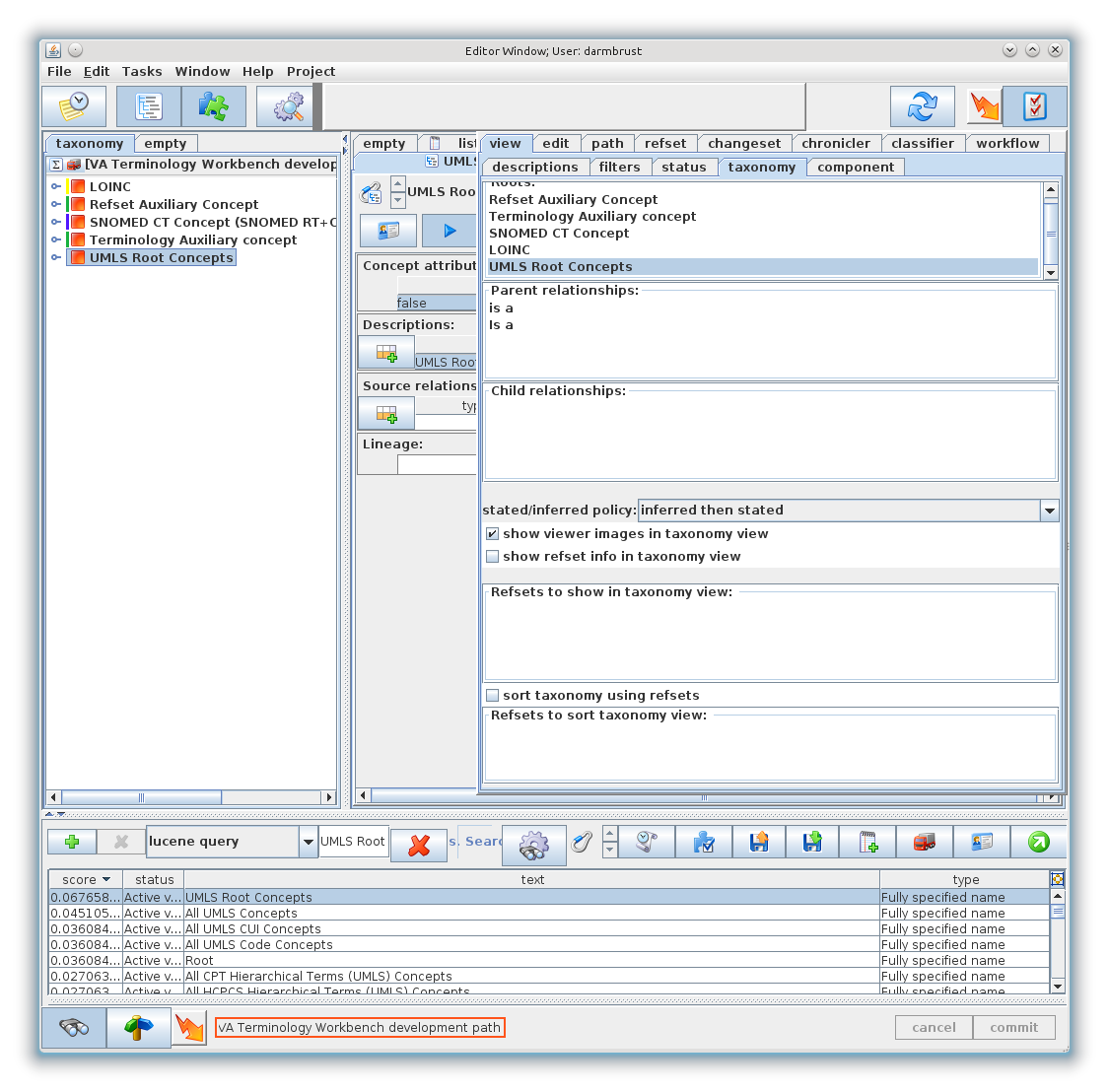


Figure – Adding UMLS Root Concepts to the Taxonomy View

Each terminology that was imported from the UMLS which has a hierarchy will have a SAB concept located under the ‘UMLS Root Concepts’ node. Under each SAB node will be the hierarchy as provided by the UMLS.

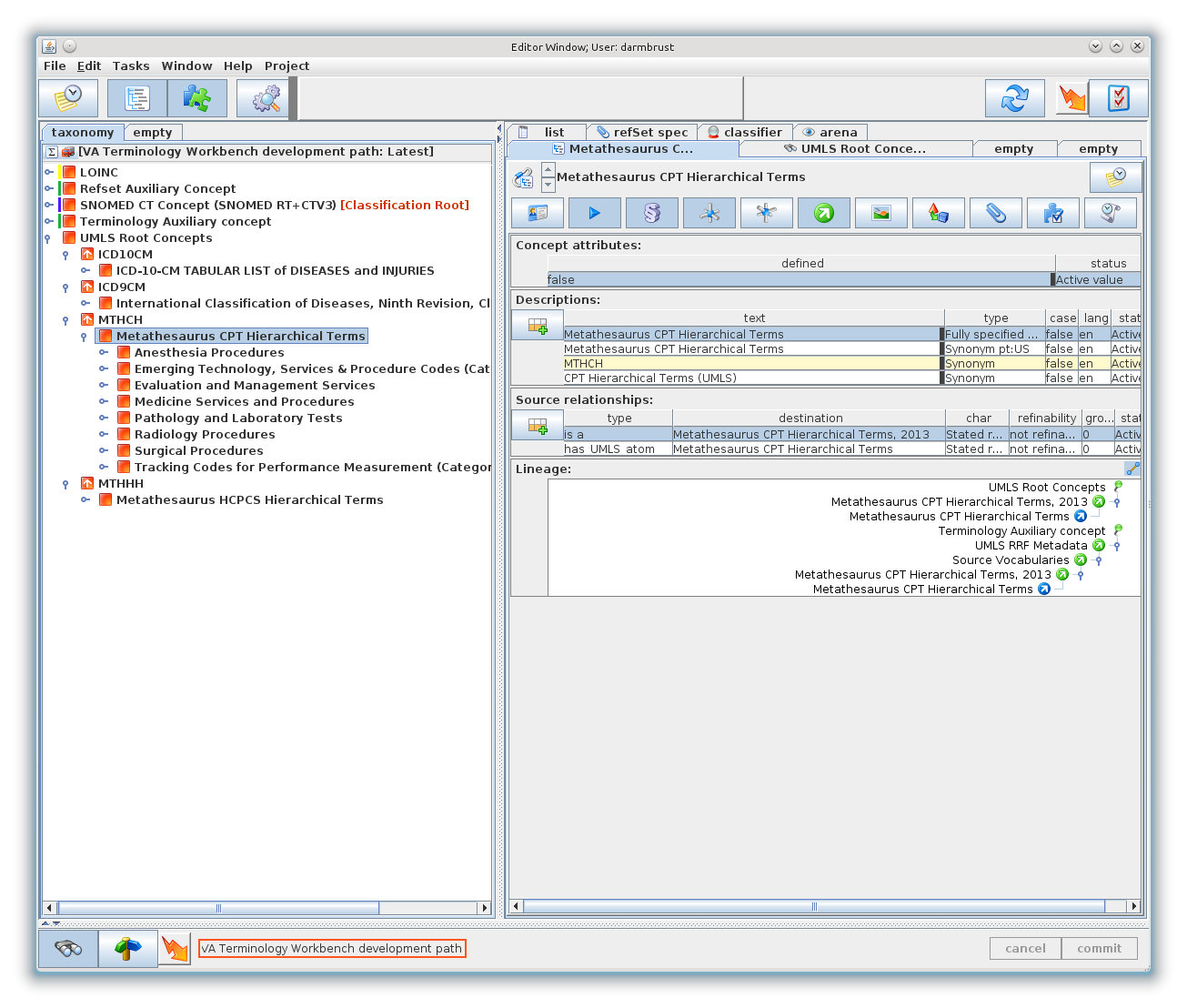


Figure – Terminology Hierarchies

The content of terminologies imported from the UMLS which do not have a hierarchy can be listed from a corresponding refset concept. See section 5.4.3 Refsets.

## UMLS Concepts in the Workbench

The UMLS contains two different types of primary identifiers – CUIs (Concept Unique Identifiers) and AUIs (Atom Unique Identifiers).

CUIs in the UMLS represent a meaning, and are linked to a number of AUI based concepts that have the same meaning.

### AUIs

In the UMLS, there is one unique AUI per description. Each AUI links to the source terminology code where the description originated. When the AUIs are loaded into the Workbench, the AUIs are grouped into unique AUI based concepts that share the same CUI, SAB, and CODE. The following table illustrates how a set of AUIs is converted into AUI based concepts in the Workbench.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **AUI** | **CUI** | **SAB** | **CODE** | **DESCRIPTION** | **WB CONCEPT** |
| A0018160 | C0000924 | MSH | D000059 | Accidents | A |
| A17996753 | C0000924 | MSH | D000059 | Accident |
| A4807043 | C0000924 | SNOMEDCT | 158001009 | Accidents NOS | B |
| A16365746 | C0000924 | SNOMEDCT | 218233008 | Unspecified accidents | C |
| A16351099 | C0000924 | SNOMEDCT | 218233008 | Unspecified accidents (morphologic abnormality) |

The unique UUID for each AUI based Workbench concept that is created is based on the combination of the CUI, SAB and CODE. Each AUI is added as an additional ID – so any AUI can be looked up directly in the WB by doing a lucene search for the specified AUI. The CODE values in the UMLS do not satisfy the requirements for an ID type in the workbench, so these are loaded as attributes.

These can be searched by doing a refset query on the “CODE” refset with a “text in member” restriction. See Figure 3 for an example of searching for the code “P00-P96”

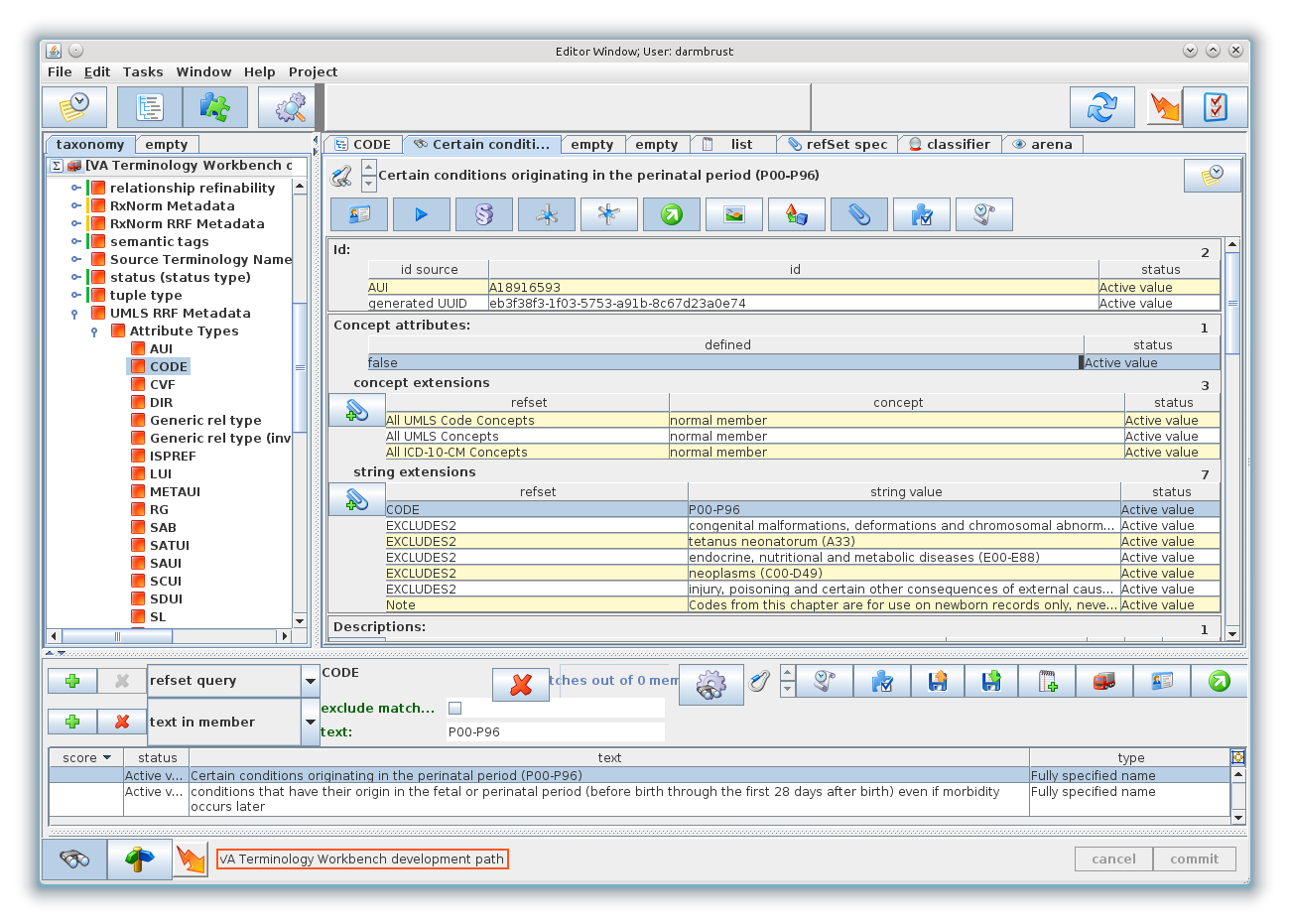


Figure - Refset Query

### AUI Based Concepts

Below, we can see the concept “Whipple’s disease” in the Workbench with a subset of attribute view toggles enabled.

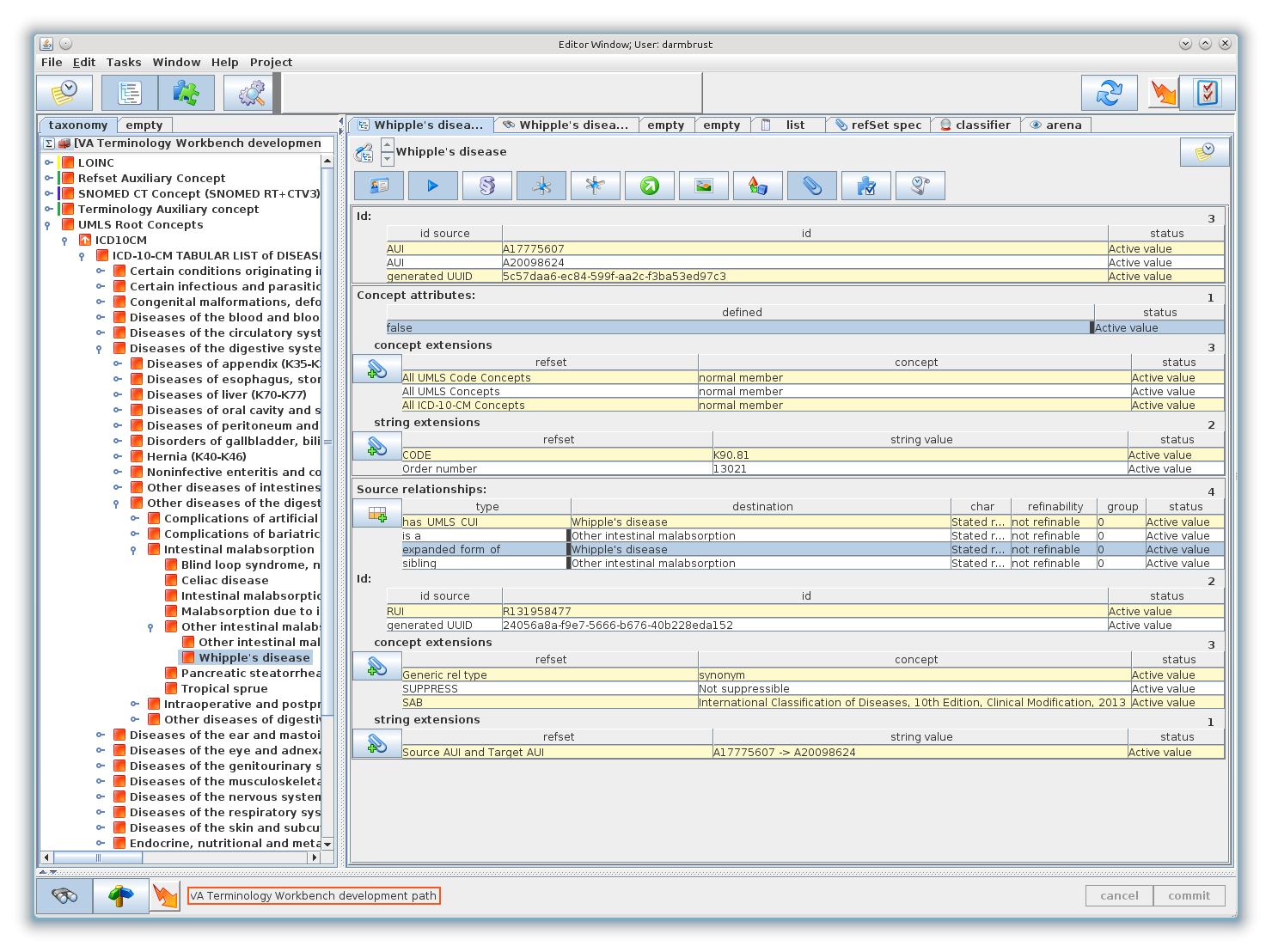


Figure - AUI Based Concept

This view shows the following information:

* Id – The Id section shows the identifiers from each AUI that was merged into this concept, and the UUID which was generated from a combination of the CUI/SAB/CODE which is required by the Workbench.
* Concept attributes – The Concept attributes section shows the current status of the Concept (active, inactive) and also any attributes and refset membership information for this concept. On this concept, there are two string extensions (CODE, Order number) and 3 refset memberships (All UMLS Code Concepts, All UMLS Concepts, All ICD-10-CM Concepts)
* Source relationships – Workbench native “is a” relationships were created from parent/child relationships in the UMLS. The UMLS relationship name is added to the “is a” relationship as concept extension. Other relationship types are used directly from the UMLS.   
    
  In the UMLS, relationships can also have attributes – each of these is included on the relationship as a concept or string extension. A “has\_UMLS\_CUI” relationship is also created on each AUI based concept, which links to the associated CUI. Other aspects of Relationships are discussed in more detail in section 5.3.

Continuing an overview of the content in the Workbench, below is the same concept with different attribute view toggles enabled.

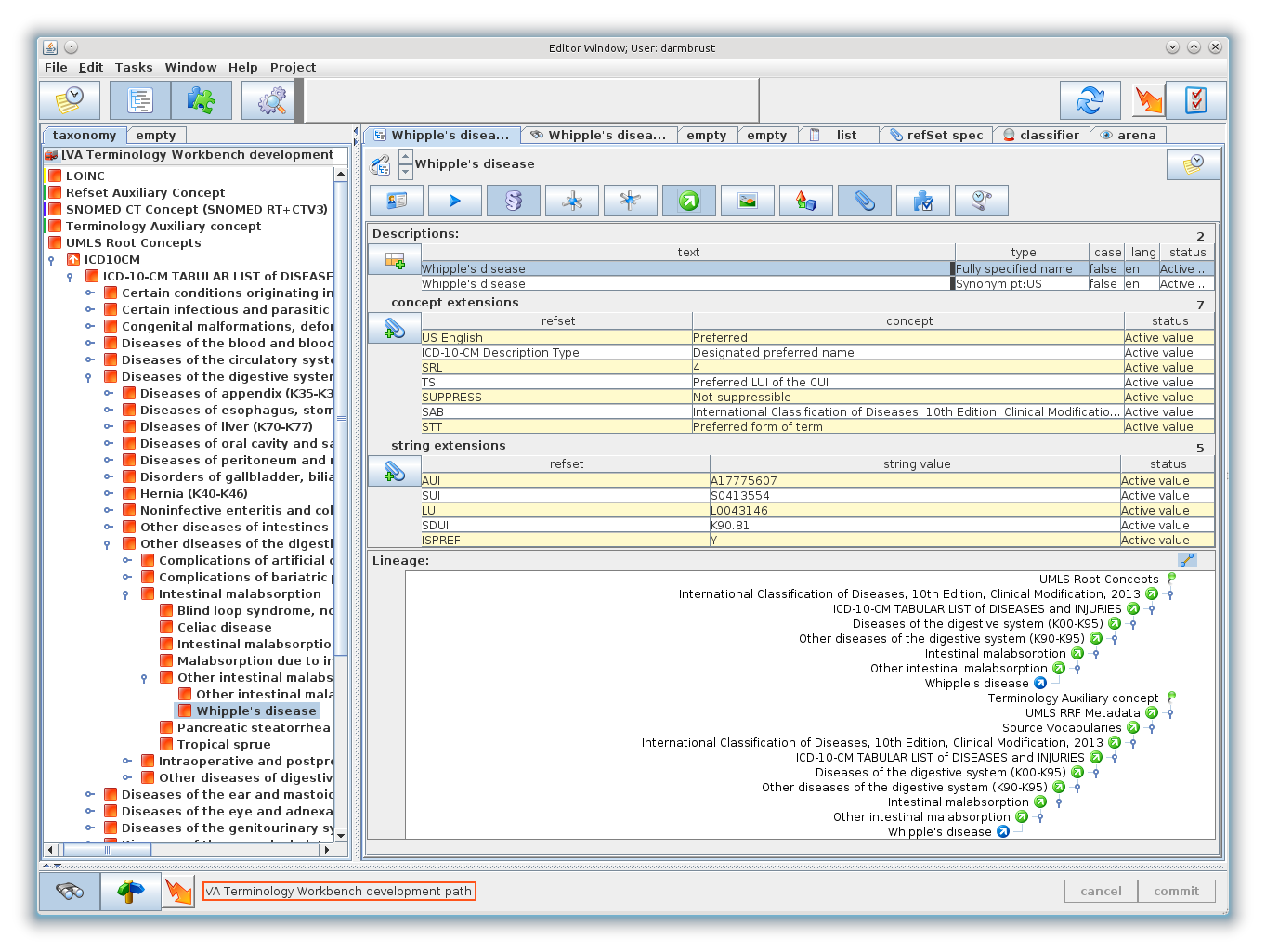


Figure - AUI Based Concept

This view shows the following information:

* Descriptions – Each string description for the current group of AUIs is created as a description on the concept. The descriptions are added using the Workbench description types (Fully Specified Name, Synonym). The UMLS description type of each description is added as an extension on each description. In this example, we show that the “Fully specified name” description came from the “ICD-10-CM Description Type” “Designated preferred name”.   
    
  Each AUI in the UMLS also has a number of attributes attached to it. These are attached to the description as concept extensions or string extensions. In general, the attribute types attached to the description correspond to the data columns in the RRF release format. The AUI identifier for the description is also attached to the description in the case where more than one AUI has been grouped on a single concept. The AUI extension is omitted in cases where there is only one AUI tied to the concept, as the AUI can then be found in the ID section.
* Lineage – The path(s) to the root concept for this concept.
  + - 1. Arena View

The primary Workbench view panels only show one level of attributes on data elements like description and relationship. In order to see nested relationships, you need to switch to the Arena View tab. Below is the same concept again, but in the Arena Viewer tab.

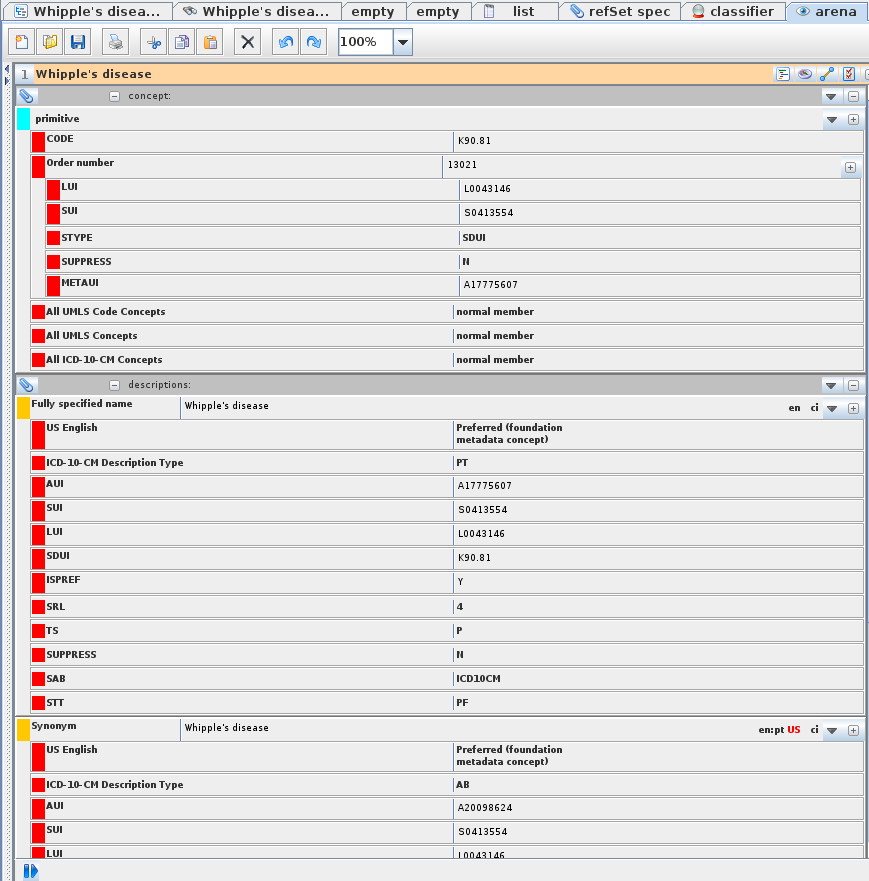


Figure - Arena Tab

Here you can see that the concept attribute “Order number” has 5 nested attributes.

### CUIs

In the Workbench, one unique concept is created for each CUI that is found in the UMLS. The highest-ranking description for the CUI concept (based on the precedence settings given during MetamorpoSys) which is available in the included terminologies is used as the FSN for this concept.

All attributes, relationships, refset memberships and descriptions which are attached to the CUI in the UMLS are attached to the CUI concept in the Workbench.

CUI concepts in the Workbench are not part of the terminology hierarchy. CUI concept codes can be looked up by ID in the lucene query panel. Figure 7 shows a search for CUI “C0023788” in the Workbench.

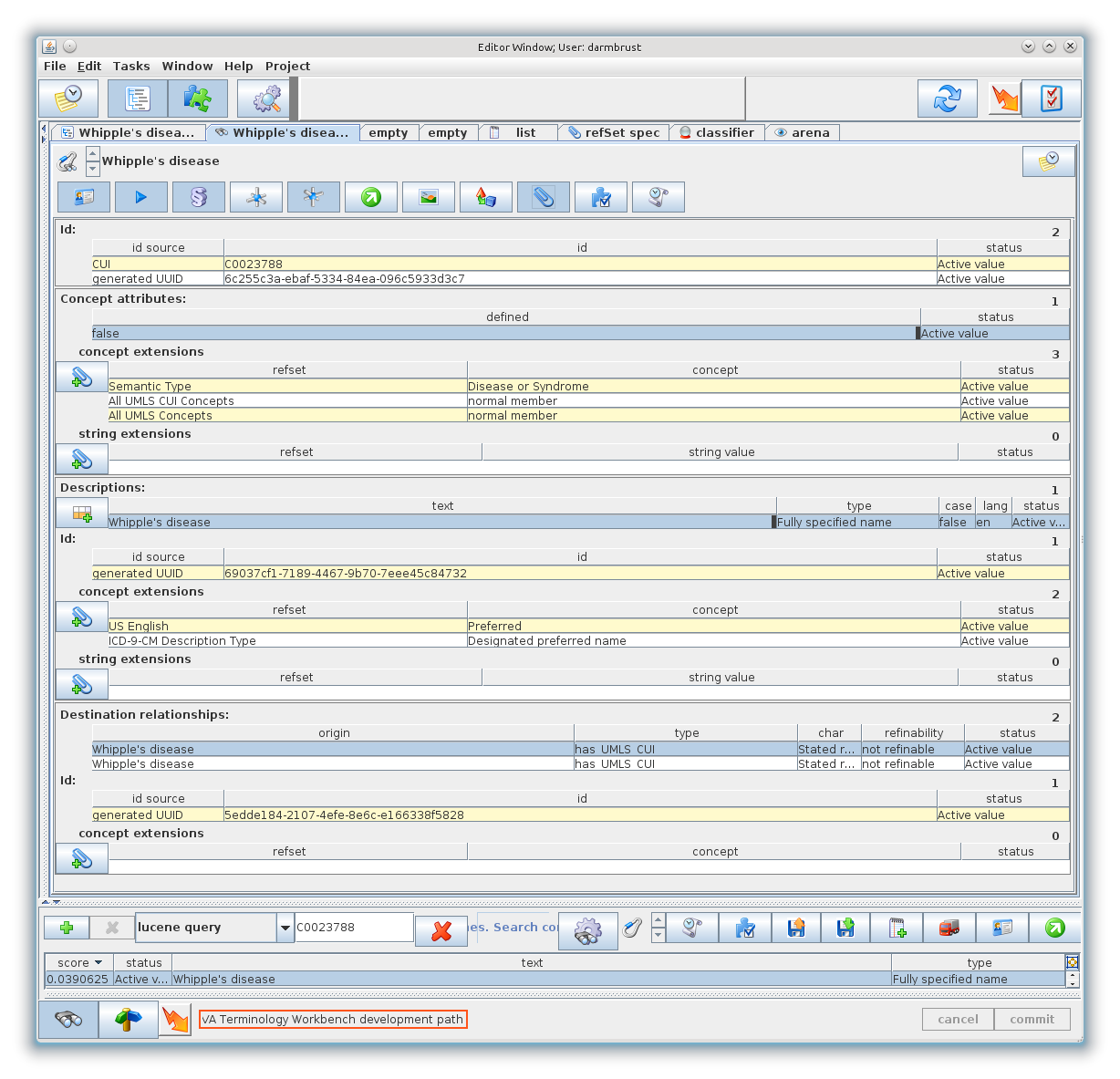


Figure - CUI Concept

On this particular CUI concept, you can see the following information:

* Id – The Id section shows the identifiers that were created from the CUI – the first is the CUI itself, and the second is the UUID which is required by the Workbench.
* Concept Attributes – The Concept Attributes section shows the current status of the Concept (active, inactive) and also any attributes and refset membership information for this concept. On this concept, there are no attributes and 3 refset memberships (Semantic Type – Disease or Syndrome, All UMLS CUI Concepts, All UMLS Concepts)
* Descriptions – A string description is created from the highest ranked AUI description that is linked to this CUI. The description is entered as the Workbench description type ‘Fully specified name’. The source of the description is added as concept extensions to each description. In this example, you can see that the “Fully specified name” description came from the “ICD-9-CM Description Type” “Designated preferred name”.
* Destination relationships – The inverse relationships that point to this CUI concept. For CUI concepts, these will include the “has\_UMLS\_CUI” relationship, which links the AUI based concepts to the CUI based concepts.

## Relationships

In the UMLS, many relationships have two different relationship names – a specific name such as “expanded form of” and a generic name such as “synonym”. When a specific relationship name is available – this is used as the Workbench relationship type. The generic relationship name is added as a relationship extension. If no specific relationship name is available, then the generic relationship name is used as the Workbench relationship type.

In the image below, we can see an example of each of these. The relationship “expanded\_form\_of” is a specific relationship type, so it has a nested “Generic rel type” attribute with a value of “SY” – also known as “Synonym”. This is the generic relationship name attached to this particular relationship in the UMLS.

The relationship “sibling” is already a generic relationship type, so it is used directly with no further qualification. There is no specific relationship name available in the UMLS for this particular relationship.

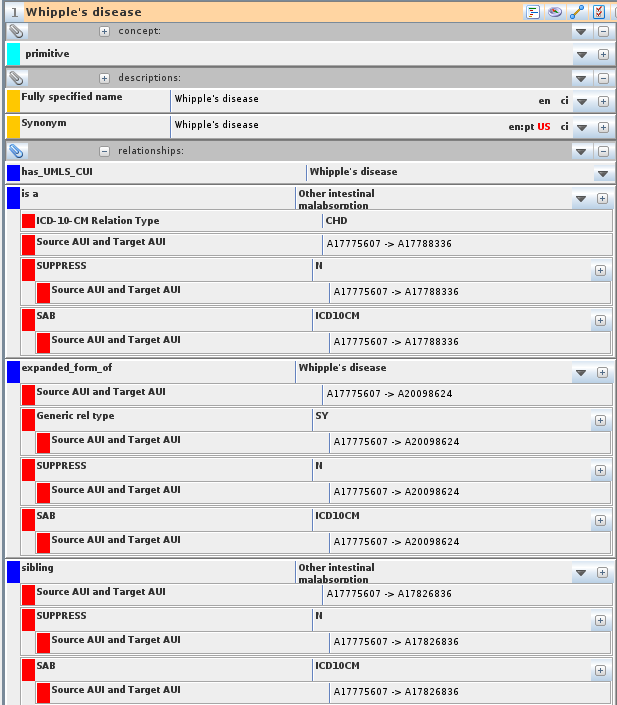


Figure - Relationship Attributes

The image also shows numerous nested attributes named “Source AUI and Target AUI”. These attributes are used to identify the specific AUI values which declared the relationship, and the nested attribute values. This is necessary because when multiple AUI’s are merged into a single concept in the Workbench it would be impossible to link back to the actual source of the relationship without this value.

Frequently, as a result of the merged AUIs duplicate definitions of the same relationships will be created. These duplicates are automatically removed during the conversion – but the Source AUI and Target AUI annotations will remain to identify each unique AUI pair that had the relationship in the UMLS. The image below shows a case where a duplicate relationship was merged.



Only one instance of the “expanded\_form\_of” relationship was created – but both AUI pairs that listed this relationship in the UMLS are added as attributes. Additionally, each relationship also declared the same “Generic rel type” attribute in this case – this duplication was removed – “Generic rel type” is only listed once, but nested attributes were added to provide a link to each unique source of the relationship information in the UMLS that specified the “Generic rel type”.

The UMLS also declares all relationships in both directions – so the UMLS source data will contain:

A –> expanded\_form\_of-> B  
B -> has\_expanded\_form -> A

The Workbench automatically supports reversing relationships – so only ½ of the relationships from the UMLS are loaded. The reverse relationships are not necessary. In this example, the relationship A –> expanded\_form\_of-> B will be loaded onto concept A – which is shown in the “Source Relationship” section for concept A. In concept B, the same relationship is visible in the “Destination Relationship” section.

The concept that defines “expanded\_form\_of” within the terminology metadata hierarchy (see section ) carries the inverse relationship name from the UMLS as a secondary description.

## UMLS Metadata

Various metadata hierarchies are created under the “Terminology Auxiliary Concept” when content is loaded from the UMLS.

### UMLS Terminology Specific Metadata

Each terminology that is loaded has a terminology specific metadata hierarchy under “Terminology Auxiliary Concept”. below, shows the metadata node for ICD-9-CM.

The metadata hierarchy for each terminology is further grouped by functional use. These groupings are described below.

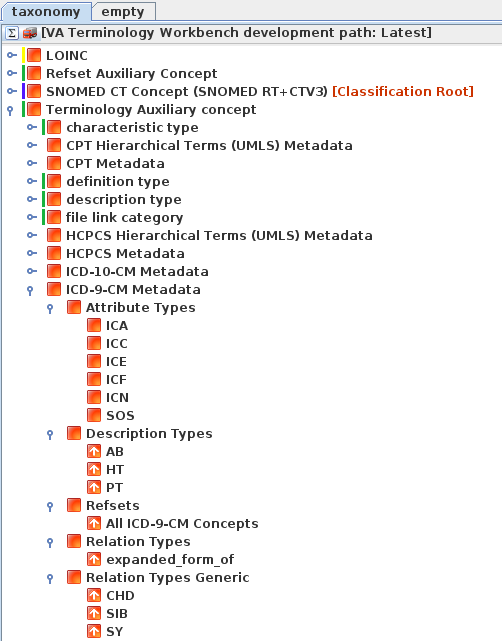


Figure - ICD-10 Metadata.

* Attribute Types – These are the extension types that are used on concepts, descriptions and relationships within the ICD-9-CM terminology. Each of these concepts also doubles as a refset, which can be dragged into the refSet spec tab – where you can see a list of the concepts that use each attribute.
* DescriptionTypes – These are the UMLS description types that are used within the ICD-9-CM terminology. These types are used as qualifying types on the core Workbench description types.
* Refsets – These are the member refsets that were created from the ICD-9-CM content. Each terminology that is loaded from the UMLS will have an “All <terminology name> Concepts” Refset. When this refset is dragged to the refSet spec tab, all AUI based concepts from the specified terminology can be viewed.
* Relation Types – These are the specific UMLS relationship types that are used within the ICD-9-CM terminology as primary relationship types.
* Relationship Types Generic – These are the generic UMLS relationship types that are used within the ICD-9-CM terminology. These types are used as both primary relationship types and as secondary attributes, depending on whether or not the UMLS relationship contains a specific relationship type.

### UMLS Shared Metadata

When data is imported from the UMLS, a “UMLS RRF Metadata” hierarchy node is also created. This hierarchy is shown in Figure 8.

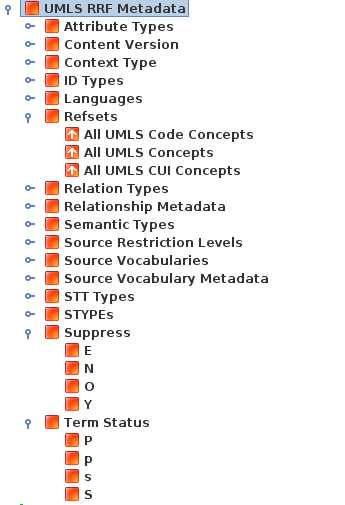


Figure - UMLS Shared Metadata

This metadata is used during the creation of Workbench attributes from various data tables and column names within the RRF format. In general, they are created to fulfil requirements of the Workbench data model, and won’t be described in detail in this document.

Of note, however, is the Refsets node – which lists 3 different member refsets which are created during a UMLS data load.

* All UMLS Code Concepts – This refset contains the AUI based concepts that were created for all terminologies that were imported from the UMLS.
* All UMLS Concepts –This refset contains both the CUI based concepts and the AUI based concepts for all terminologies that were imported from the UMLS.
* All UMLS CUI Concepts – This refset contains all of the CUI based concepts that were created for all terminologies that were imported from the UMLS.

### Refsets

Refsets are also stored under the Workbench hierarchy “Refset Auxiliary Concept” -> “refset identity” -> “VA Refsets“.

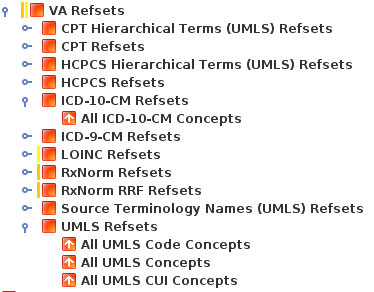


Figure – Refsets

Here, you can see the member refset concepts for each individual terminology, in addition to the UMLS shared refsets.

## Workbench Data Mapping

The release and version information of the UMLS data is attached to the root node “UMLS Root Concepts” as a set of string annotations.

The rest of the data from a UMLS release is transferred into the workbench according to the table below.

|  |  |
| --- | --- |
| **RRF / UMLS Data File** | **Workbench Data Type** |
| MRCOLS | This is data about the UMLS release itself – it is used internally in the converter, but not added to the WB. |
| MRCONSO | This file contains all of the descriptions within the UMLS, and a number of other attributes. The descriptions are loaded onto the appropriate WB concept with a type of “Fully Specified Name” or “Synonym”, depending on the precedence settings and the descriptions available on a particular concept. The UMLS description type is added to each description as an annotation. Other attributes from this table are also attached to the description as extensions on the description.  A CUI concept is also created for each unique CUI concept found in this file. |
| MRDEF | This file contains Definitions and a few attributes that attach to the definition. These are loaded into the Workbench on the appropriate concept as descriptions of type Definition. Attributes from this file are added as annotations on each definition. |
| MRDOC | This is data about the attribute types used in the UMLS release itself – it is used internally in the converter to document the various metadata concepts that are created – but in general is not captured directly into the WB. |
| MRFILES | This is data about the UMLS release itself – it is used internally in the converter, but not added to the WB. |
| MRHIER | This file is used to locate the root concept of each terminology that is loaded from the UMLS. The rest of this data duplicates relationship information that we read from MRREL, so it is not loaded. |
| MRRANK | This file contains the precedence information used to select which description type should become the Fully Specified Name attribute for each WB Concept. |
| MRREL | Relationships between AUIs and CUIS. Relationships are processed to remove duplicates, and combine inverse relationships into a single relationship – and are then added to the appropriate Workbench Concept. Attributes from this file are added to the relationship as annotations. |
| MRSAB | This file contains information about each terminology that is available in the UMLS. The relevant data (depending on the terminologies being loaded) is added to the root node for that terminology as string extensions on the concept. |
| MRSAT | This file contains attributes which may be attached to CUIs, AUIs, or relationships. The attribute types from this file are used to create metadata concepts, and the attribute values are attached to the appropriate workbench component – Concept, Relationship, etc. This file also contains annotations on each attribute – each of these is added as an annotation on the Workbench Attribute object that is created for the attribute value. |
| MRSTY | This file contains the Semantic Type information for the CUIs. This data is added to each CUI based concept as a member refset reference. |
| Other RRF Data files | Not needed at this time for the conversion into the Workbench. |

# Appendix A

These are the type mapping statistics from the 2013AA release of UMLS, for the requested terminology subset.

Due to the tremendous amount of individual attribute mappings and counts, the data presented here is a high level summary of the statistics from the current release.

The full in-depth statistics are available within the UMLS-econcept maven artifact which is published in step 6 of the Converting the source content step, documented above.

|  |  |
| --- | --- |
| Data Type | Loaded into Workbench |
| Total Concepts | **313242** |
| Additional Concept Identifiers | 432707 |
|  |  |
| Total Relationships | **1063521** |
|  |  |
| Total Descriptions | **432707** |
| FSN | 313242 |
| Synonym | 119465 |
| Description Annotations | 3694015 |
|  |  |
| Total Annotations | **10152908** |
| Concept Annotations | 507001 |
| Other Annotations | 9645907 |
|  |  |
| Total Refset Members | **790642** |
| All CPT Concepts | 32444 |
| All CPT Hierarchical Terms (UMLS) Concepts | 1041 |
| All HCPCS Concepts | 5952 |
| All HCPCS Hierarchical Terms (UMLS) Concepts | 347 |
| All ICD-10-CM Concepts | 101954 |
| All ICD-9-CM Concepts | 22401 |
| All Source Terminology Names (UMLS) Concepts | 19 |
| All UMLS CUI Concepts | 149084 |
| All UMLS Code Concepts | 164158 |
| All UMLS Concepts | 313242 |

1. http://www.ncbi.nlm.nih.gov/books/NBK9685/ [↑](#footnote-ref-1)