nehta

SNOMED CT-AU 20131130 Release
Reference Set Implementation Toolkit Guide

11 November 2013

Approved for external information

National E-Health Transition Authority

National E-Health Transition Authority Ltd
Level 25
56 Pitt Street
Sydney, NSW, 2000
Australia
www.nehta.gov.au

Disclaimer

The National E-Health Transition Authority Ltd (NEHTA) makes the information and other material ("Information") in this document available in good faith but without any representation or warranty as to its accuracy or completeness. NEHTA cannot accept any responsibility for the consequences of any use of the Information. As the Information is of a general nature only, it is up to any person using or relying on the Information to ensure that it is accurate, complete and suitable for the circumstances of its use.

Document control

This document is maintained in electronic form and is uncontrolled in printed form. It is the responsibility of the user to verify that this copy is the latest revision.

Document information

Key information

Owner	National Clinical Terminology and Information Service, NEHTA
Date of next review	14 April 2014
Contact for enquiries	NEHTA Help Centre t: 1300 901 001 e: help@nehta.gov.au

Quality reviews

dv #	Reviewer(s)	Type of review	Purpose
002	Terminology Analyst, Clinical Terminology Lead	Content Review	Confirmed accuracy of Content Update
003	Technical Writer	Editorial Review	To ensure that the document meets NEHTA's editorial standards

Product version history

Product version	Date	Release comments
20080829	29 Aug 2008	Initial Release
20090416	16 Apr 2009	Revised to include updated definitions and scenarios for implementation.
20091130	30 Nov 2009	Updated for SNOMED CT-AU Release 1.0
20100531	31 May 2010	Updated for SNOMED CT-AU May 2010 Release
20100731	31 Jul 2010	Publication of Emergency department reference sets
20101130	30 Nov 2010	Updated for SNOMED CT-AU November 2010 Release
20110531	31 May 2011	Updated for SNOMED CT-AU May 2011 Release
20120531	31 May 2012	Updated for SNOMED CT-AU May 2012 Release
20121130	30 Nov 2012	Updated for SNOMED CT-AU November 2012 Release
20130531	31 May 2013	Updated for SNOMED CT-AU May 2013 Release

Table of contents

1 Introduction

This guide has been created as a short introduction, or “quick start” reference, for developers wishing to work with SNOMED CT-AU[1] reference sets using the supplied application.

Caution: This guide and the associated application are provided for the purposes of demonstrating one way to implement SNOMED CT-AU Release Format 2 (RF2) files. It is not intended for clinical use.

1.1 Purpose

The purpose of this document is to provide developers with a basic introduction to SNOMED CT-AU (RF2) files using the Refset guidance sample application. This document provides information on setting up and using this application on Java and .NET platforms. The application is provided to demonstrate one example of:

1. Loading the SNOMED CT-AU content from the Terminology Bundle (SNOMED CT RF2) into a relational database.
2. Building and executing either a Java or .NET application to perform some simple query operations to obtain information from the SNOMED CT-AU content.

This document details:

1. The requirements/pre-requisites for running the application.
2. How to install the application.
3. Basic use of the application.
4. Some of the reasoning in the design applied to the application.

1.2 Intended audience

The audience for this document is software developers who want to further understand the format and relationships of the SNOMED CT-AU RF2 data files. Although the files are provided in RF2, there are some deviations from the International Health Terminology Standards Development Organisation (IHTSDO) format. These are detailed in the *Deviation From RF2 Specifications* document, which is produced with every SNOMED CT-AU release and is available on the NCTIS Secure Website to licence holders.

1.3 Scope

This document provides information for installing and using the provided sample application on Microsoft Windows®, Mac OS® and Linux platforms.

It does not provide any information regarding the installation or configuration of pre-requisite software required by the sample application.

It does not intend to provide a detailed description of SNOMED CT-AU, but rather to demonstrate one simple way to utilise it.

[1] IHTSDO®, SNOMED® and SNOMED CT® are registered trademarks of the International Health Terminology Standards Development Organisation.

1.4 Overview

Content provided in the SNOMED CT-AU Terminology is a collection of simple tab-delimited text files; essentially a database exported to file.

Implementers of SNOMED CT-AU may wish to load the RF2 file into a relational database as a source of terminology content for their own systems. The sample application demonstrates this by defining a schema for the database and then loading the content of the RF2 files into this schema.

The sample application also demonstrates the querying of the data through the following simple functionality:

1 Display the details of a concept. The SNOMED CT identification number SNOMED CT-AU and the active descriptions (or “terms”) are retrieved.

2 Identify the Australian Dialect (en-AU) acceptability of a description, being either “preferred” or “acceptable”, by looking up the description in the Australian Dialect Reference Set (ADRS).

3 Specify all the Clinical and Foundation reference sets that a concept is an active member of.

4 Specify all the concepts that are active members of a specific Clinical or Foundation reference set.

The application provides the following operations for retrieving concepts:

1 Find a single concept by its SNOMED CT identifier.

2 Find concepts with an active description containing a textual search term.

3 List all the members of reference set by providing the SNOMED CT identifier of the reference set.

Caution: Please be aware that this is only ONE method of consuming SNOMED CT-AU data; there are numerous ways of doing this, depending on the objective of the implementation.

1.5 SNOMED CT—AU Reference set implementation bundle

The SNOMED CT-AU Reference set implementation bundle contains the following files:

1 doc2_RefersetImplementationToolkitGuide_AU1000036. It provides information for both the .NET and Java applications.

2 tls2_RefsetSampleAppNet_AU1000036 – The .NET version of the application.

3 tls2_RefsetSampleAppJava_Au1000036 – The Java version of the application.

1.6 Questions and feedback

The NCTIS’s products and development relies on the input and co-operation of the healthcare community. We value your feedback and encourage questions, comments or suggestions about our products. Please direct your correspondence to: help@nehta.gov.au.

2 SNOMED CT-AU overview

2.1 Release content

SNOMED CT-AU is released as a group of five main files.

- Terminology bundle:
  - ZIP file containing terminology data files and documentation
  - Uses IHTSDO file naming convention within the ZIP file
- Release note:
- Mac OSX Viewer – read only IHTSDO Workbench
- Windows Viewer – read only IHTSDO Workbench
- MD5 checksum file

As stated, the ZIP file's content is named using the IHTSDO's file naming conventions for consistency, and separates:

- Core content – Release Format 1 (RF1) and RF2
- Reference sets (RF2 format)
- Documentation

There is also a release note which describes the content of the release, any known issues and a summary of the changes.

Supporting the release are two versions of the IHTSDO Workbench preloaded with the release data, one for the Macintosh operating system, and the other for Windows. This enables consumers to browse the data without having to load it into a system, and is provided as a convenience – it is not intended for implementation.

Finally there is an MD5 checksum file which can be used to ensure that the terminology bundle was downloaded without error.

2.2 RF1 and RF2 specification

RF2 is a specification published publically on the IHTSDO website: http://www.ihtsdo.org/publications/tools-for-terminology-developmentmaintenance/

It consists of:

- A data structure specification for the "core" tables – concepts, descriptions, relationships and identifiers.
- A reference set specification which covers reference set data structures, and the many patterns and applications.
- An update guide which describes the reasons for creating RF2, key changes in RF2 and principles which guided its creation.

RF2 is superseding RF1, and as such has been adopted for both SNOMED CT-AU and the sample application provided with this document. RF1 releases are to be discontinued in the future in favor of RF2 at both national and international levels.

RF1 has no true specification; rather its format is documented in the Technical Implementation Guide (TIG) and Technical Reference Guide (TRG) that has been published with every SNOMED CT release.

Both RF1 and RF2 are tab delimited text files and are essentially relational. Therefore, both may be loaded into a relational database and queried/mined, which is the way most developers interact with the content. This allows data to be picked out and transformed into a form that best suits the implementation. It is rare that systems require all the data represented in the release files, as few implementations support the full set of use cases for which SNOMED CT and the release formats are intended.

SNOMED CT (RF1 or RF2) may also be implemented into relational or object/associative databases using a variety of schemas. These decisions must be made based on the most appropriate design for the intended implementation and can only be made by those developing the required system.

Yet another way would be to use a terminology or ontology server. This allows systems to decouple the terminology implementation from the rest of the application, which often have different change lifecycles. This is also particularly attractive for enterprises that can connect many systems to a central terminology server, making it easier to manage terminology versions in use across multiple systems. However this approach may not suit all implementations as systems become reliant on access to a terminology server. There are a number of commercial terminology servers available, and usually they serve multiple classifications and terminologies, not just SNOMED CT.

Key points to remember are:

- Both RF1 and RF2 are delivered as tab-delimited text files.
- They can be loaded into a relational database and queried/mined – this is what the sample application does.
- They can be implemented in many ways:
  - Relational databases
  - Object/associative databases
  - Terminology/ontology servers
- There is no one size fits all solution – the solution will depend upon the nature of the implementation.

2.3 Files and updating

When planning to take updates, RF2 offers “Full”, “Snapshot” and “Delta” types of release[2].

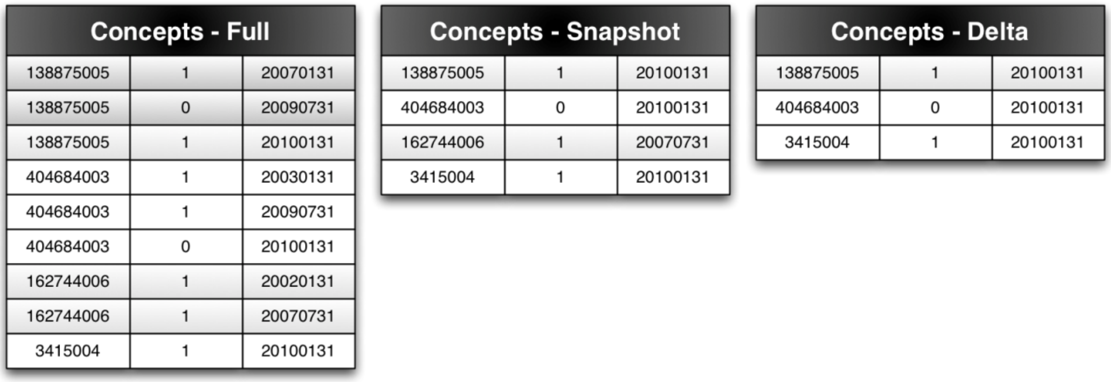
As SNOMED CT has versioning built into its release format, RF2 can be distributed in three forms – Full, Snapshot and Delta. The descriptions for these are as follows:

A Full release contains every version of every component (concept, description, relationship, reference set member) ever released.

A Snapshot release contains the single latest version of every component ever released. This is the same as RF1 releases currently produce.

[2] SNOMED CT-AU currently releases only in RF2 Snapshot format, although this may be extended to include Full and/or Delta in the future.

A Delta release contains only the new versions of components that were changed since the last publication. The figure below depicts the differences between the release types.



Concepts - Full

138875005	1	20070131
138875005	0	20090731
138875005	1	20100131
404684003	1	20030131
404684003	1	20090731
404684003	0	20100131
162744006	1	20020131
162744006	1	20070731
3415004	1	20100131

Concepts - Snapshot

138875005	1	20100131
404684003	0	20100131
162744006	1	20070731
3415004	1	20100131

Concepts - Delta

138875005	1	20100131
404684003	0	20100131
3415004	1	20100131

Figure 1: Breakdown of the different release types

This figure shows for each release type an identifier, active/inactive indicator (1 or 0), and an effective time in the format yyyymmdd for an example version history of components. From this example it is clear that the:

- Full release contains multiple versions of each component each with a different effective time.
- Snapshot contains only the most recent version of each component from the Full set.
- Delta contains only versions of components changed in a release – in this case the 20100131 release. A Delta release is only effective if ALL previous releases have been consumed.

These release types can be used in different ways to maintain and upgrade terminology in systems. For example a system may take “Snapshot” releases, dropping data and reloading the new “Snapshot” after every release, while another system may take an initial “Snapshot” release and then apply “Delta” releases like patches for each subsequent release. Again this depends upon the application, setting and constraints, so the release types are aimed at providing options.

2.4 Implementation considerations

When planning to implement a system to consume SNOMED CT-AU data, it is important to spend sufficient time planning your objectives, and research the best way to achieve them. Correct planning will minimise risks and issues once the system has been implemented.

One major consideration when implementing with SNOMED CT or any terminology is maintenance. Current license obligations (Section 6.2 of the *Australian National Terminology Release Licence Agreement*[3]) require you to be licensed to updates to new terminology versions within 180 days of publication. This is quite achievable in

[3] Available to all licensees on the NCTIS Website at https://nehta.org.au/aht/index.php?option=com_docman&task=doc_download&gid=327&Itemid=40.

well-planned and built systems, but can pose problems if not considered prior to implementation.

Licensing obligations aside, consuming regular updates has other benefits including:

- Users of affected systems use more up to date terminology.
- When interoperating with other systems, using the most recent terminology version will limit interoperability issues.

As mentioned, system design can have a profound effect on the cost of regular terminology updates and must be considered to ensure that terminology can be updated regularly with the minimum of effort and impact. There are many strategies to this, and their applicability depends upon the implementation. However in general they are:

- Store the full terminology history over time and, and use queries or views that are aware of this history. In this scenario it is possible to append data to the tables when new terminology is delivered, for example a series of Delta files.
- Store a snapshot of the terminology at a point in time, much like a “Snapshot” release. It is necessary to drop and reload the data when a new terminology release is consumed, however the storage requirements are lower and is therefore smaller and faster.
- Use a terminology server that handles terminology versioning and upgrades.

Other things to be aware of are:

- Refset identifiers (an identifier that identifies a member of a reference set) are UUIDs[4]. UUIDs are 128 bit numbers, and pose problems for some relational databases.
- Indexing must be considered based on the schema (or equivalent) for the data store, and must be driven by the scenarios the implementation must deal with. However, creation of a transitive closure is worth considering to pre-compute inherited relationships. This generates a large amount of data; however it provides fast query times in relational databases rather than attempting to execute hierarchical queries.

It is also worth considering that the terminology data is not really relational, even though it is distributed in files that use a relational format. SNOMED CT is more truly a graph of nodes (concepts) connected by relationships of different types, and each with different meanings. If the implementation needs to extensively use the relationships in SNOMED CT, and treat the data as a graph rather than simplistically as a list of codes and values, a traditional relational database may not be a good fit.

Further information is available through the information and links in Appendix A.

2.5 SNOMED CT data – example

The following diagram is an example of two concepts, the first concept is fully defined and the second is not fully defined:

[4] SNOMED CT-AU currently uses SNOMED CT Identifiers for reference set member identifiers as it was implemented based on an earlier version of the RF2 specification. The latest RF2 specification states that the reference set member identifier is a UUID and SNOMED CT-AU will ultimately change to match this.

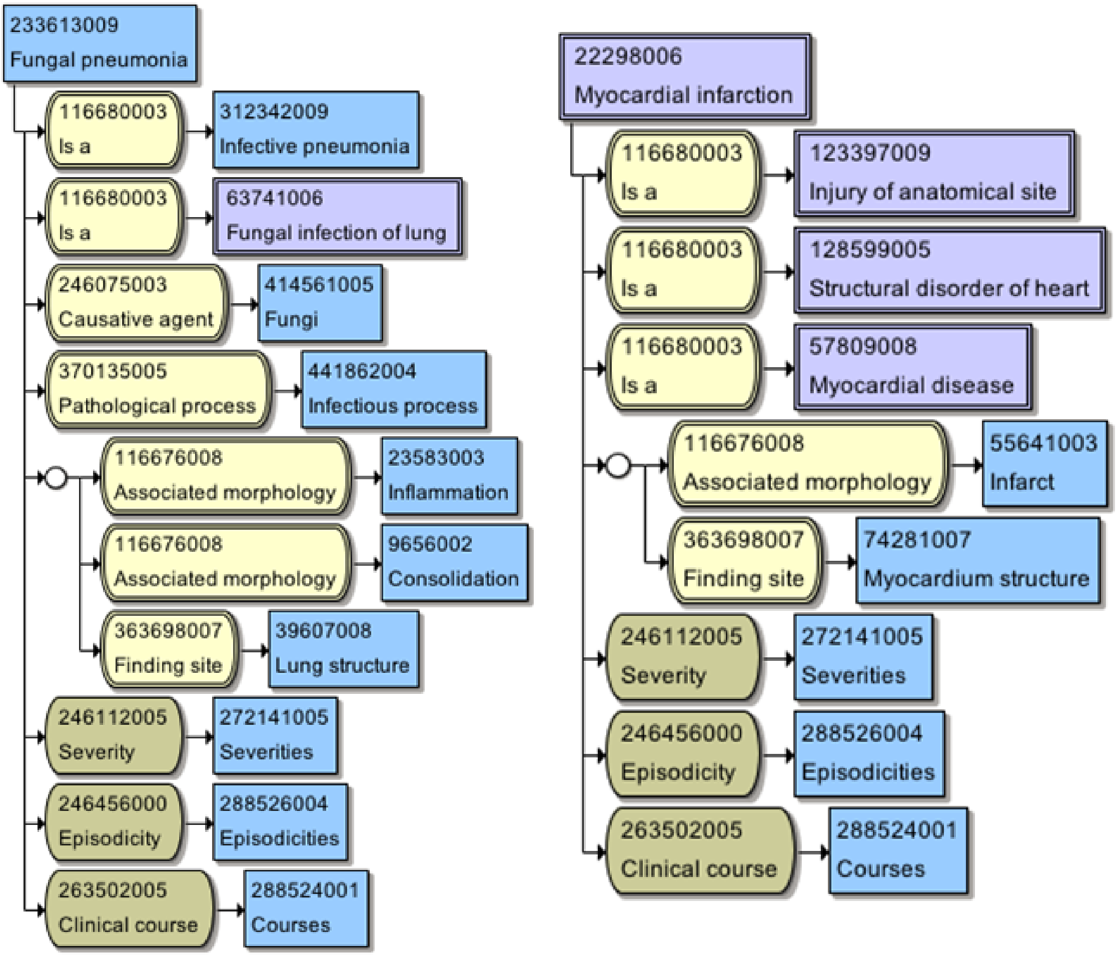


Figure 2: SNOMED CT data example

The intent of the diagram is to show that concepts have many relationships of different types in order to model meaning. The circles in the diagram represent a “role group” which is used to indicate a set of relationships that must be interpreted as a group rather than individually. This means for the “fungal pneumonia” concept that the inflammation and consolidation occur in the lung structure. This is particularly important when, as an example, a concept has two different morphologies at two different sites – the role grouping specifies which one occurs at which site.

When “flattening” a reference set to a list of concept identifiers and descriptions, as the sample application does, this richer information is lost. However, this may still be a valid approach depending upon the needs of the application.

2.6 Basic data model

This following diagram is of a data model that shows (simplistically):

- Concepts have multiple descriptions:
  - these descriptions may be of different types.
- Concepts also have multiple relationships to other concepts:
  - again these may be of different types.
- Concepts, descriptions and relationships may have alternate identifiers managed by the identifiers table.

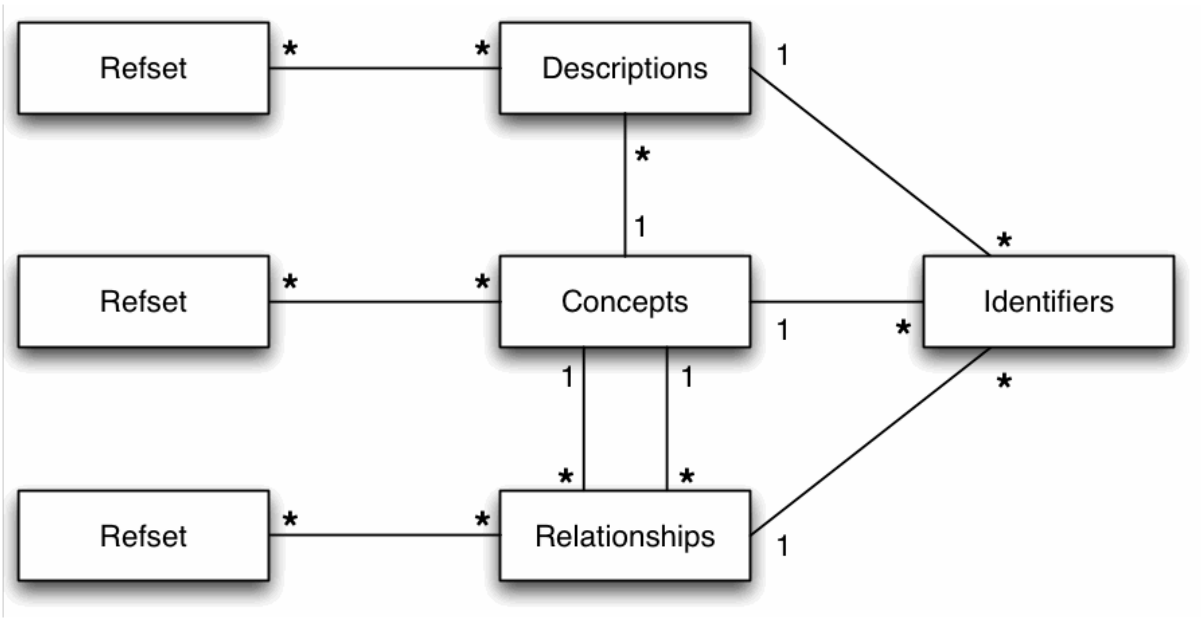


Figure 3: Basic data model

Additionally worth noting:

- Reference sets are identified by a concept in the meta-data hierarchy (see children of the concept 900000000000455006 |*Reference set*|) – its identifier becomes the Refset ID.
- Reference set rows are identified by a UUID, and depending upon the pattern, may have a number of additional annotation columns.

3 Sample application design

3.1 Database schema definition

The database schema defined in the sample application comprises of six tables. The following four tables model the standard SNOMED CT core components:

1. Concepts
2. Identifiers
3. Relationships
4. Descriptions

In addition, the following tables have been defined to model reference sets:

1. Concept reference set:

   This table will contain the Clinical and Foundation reference sets. These entries reference a concept component and define a concept attribute value for the membership.

2. Description reference set:

   This table will contain the Australian Dialect reference set (ADRS). These entries reference a description component and define a concept attribute value for the membership.

Note: There is no “right” or “wrong” schema for these files; it is dependent on the way the data is intended to be used.

The schema (or DDL) defined in the sample has been chosen purely to fulfil the operations performed by the application. It may, for example, be preferable to an implementation of SNOMED CT-AU to model an individual reference set in its own table. Another alternative may be to create multiple tables for each component with each table containing a particular version (effectiveTime) of the content in order to identify changes between SNOMED CT-AU releases.

3.2 Identifiers

Identifiers are a component of the SNOMED CT core and must be included in the database schema and data load operations. They are not currently utilised by the functions used in the sample application.

3.3 Entity-relationship diagram

The diagram below shows the schema used in the sample application:

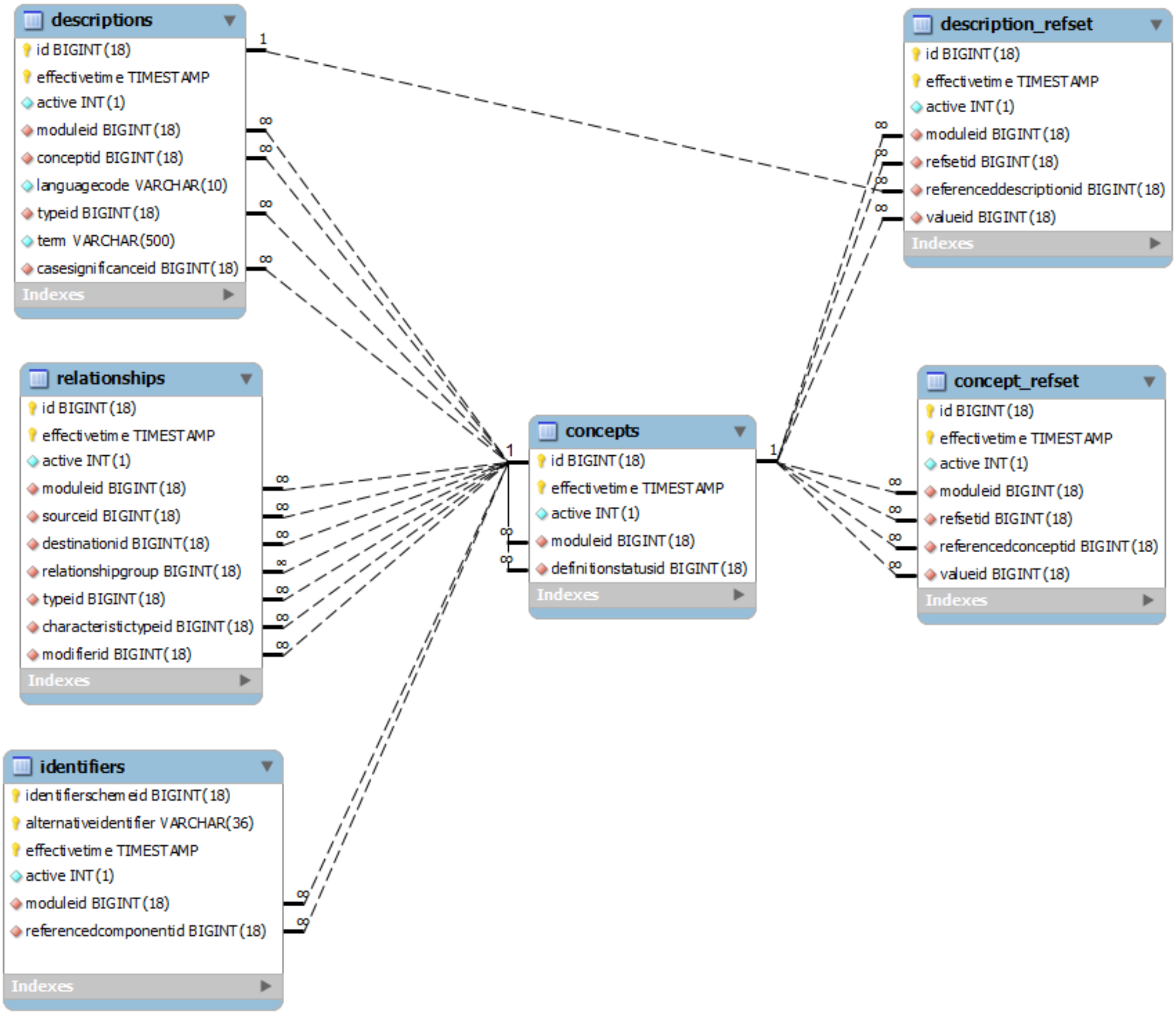


Figure 4: Database schema

3.4 Database design notes

There are some elements of design rationale in the sample application that should be observed:

1 MySQL

The DDL and SQL queries used in the application have been built and tested for connecting to a MySQL database. This product was selected as a free and easily available choice for a relational database.

2 No referential integrity

Foreign key constraints have been specified in the DDL to define the relationships between tables in the schema, however at the same time the use of MyISAM database engine has been specified. MyISAM will ignore these constraints and does not provide referential integrity checking or transaction support. Due to nature of the SQL select queries used in the sample application these features are not required and so it was deemed acceptable to use MyISAM which provides a significant performance advantage over other engines, particularly during the data load process.

3 Indexing

Only indexes useful to the SQL queries executed by the application have been created.

4 No versioning

The sample application has been created to load in a SNAPSHOT release of the SNOMED CT-AU Terminology Bundle. This type of release contains only the latest version/state of each component. It does not contain the full history of changes made to a component. The application builds the database by removing all previously loaded data/content. Owing to the use of a SNAPSHOT release the SQL queries do not need to evaluate the effectiveTime field to determine the latest version of a component.

3.5 Code structure

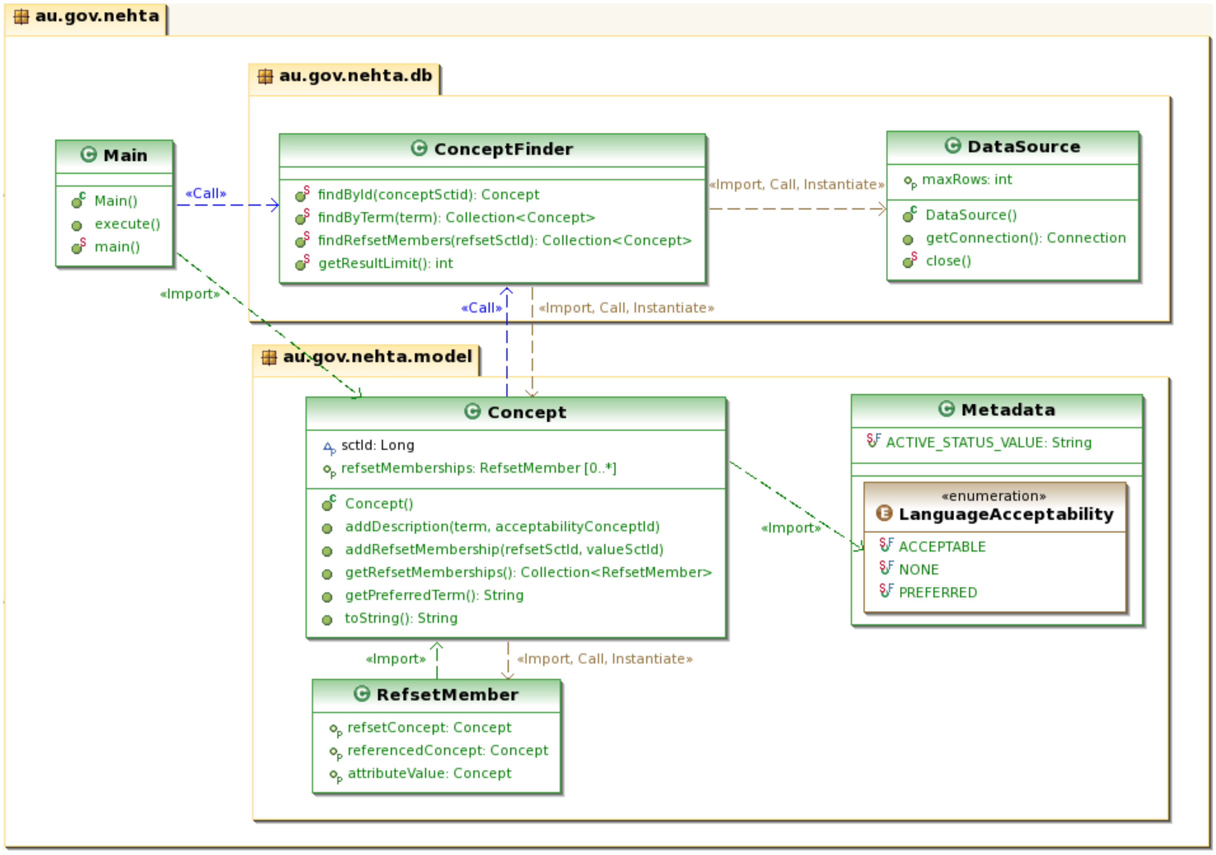

au.gov.nehta
au.gov.nehta.db
Main
Main()
execute()
main()
«Call»
«Import»
ConceptFinder
findById(conceptSctid): Concept
findByTerm(term): Collection<Concept>
findRefsetMembers(refsetSctId): Collection<Concept>
getResultLimit(): int
«Import, Call, Instantiate»
DataSource
maxRows: int
DataSource()
getConnection(): Connection
close()
«Call»
«Import, Call, Instantiate»
au.gov.nehta.model
Concept
sctId: Long
refsetMemberships: RefsetMember [0..*]
Concept()
addDescription(term, acceptabilityConceptId)
addRefsetMembership(refsetSctId, valueSctId)
getRefsetMemberships(): Collection<RefsetMember>
getPreferredTerm(): String
toString(): String
«Import»
Metadata
ACTIVE_STATUS_VALUE: String
«enumeration»
LanguageAcceptability
ACCEPTABLE
NONE
PREFERRED
«Import»
«Import, Call, Instantiate»
RefsetMember
refsetConcept: Concept
referencedConcept: Concept
attributeValue: Concept

Figure 5: UML class diagram of Java source code

Table 1: Class names and descriptions

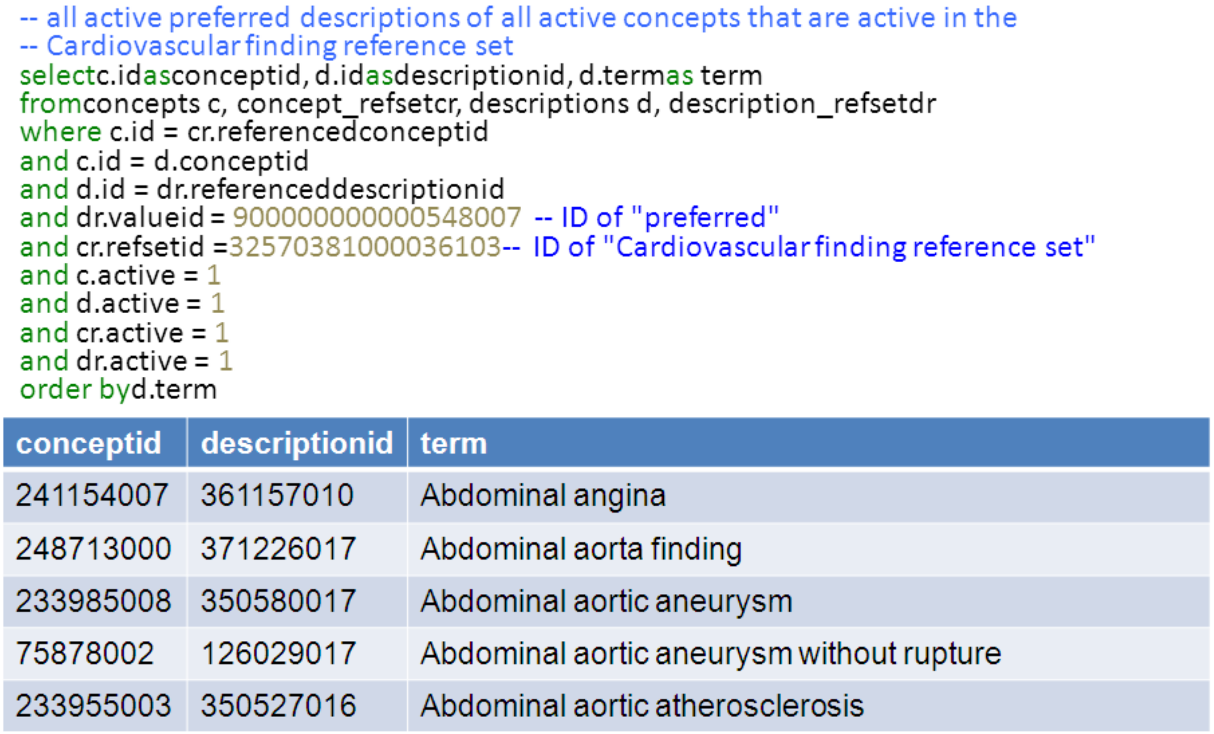
Class name	**Description**
Main	The main class of the application to be executed. Provides a simple command line interface to perform SNOMED CT-AU query operations.
DataSource	Maintains the single connection to the database. The configuration to establish the connection may be declared in the properties file *application.properties*. Alternatively a different file may be specified by setting the system property *config.file*. These properties will be loaded in as standard system properties. If the specified properties file cannot be found the application will continue without it (no errors reported).
ConceptFinder	Find concepts for given search parameters using the DataSource.
Concept	A data object modelling the "Concept" component in SNOMED CT. It is not comprehensive and contains just the elements utilised by this sample application. It contains the descriptions for the concept and, for each description; its acceptability as defined in the Australian Dialect Reference Set (ADRS). It also contains each reference set membership this concept participates in.
RefsetMember	A simple Concept Attribute Value reference set member. This class only captures the following elements of the membership: • The reference set • The referenced concept – the concept belong to the reference set • The attribute value – a concept value attached to the membership
Metadata	Defines SNOMED CT and SNOMED CT-AU Metadata (constants) utilised by the application.

Please see *javadoc* in the source code for more information.

3.6 Querying example

For the purpose of the example, the following output is assumed to be the result of a query made against the example schema.

The following query shows all the active preferred descriptions and their concept and description ids for the Cardiovascular finding reference set:



```
-- all active preferred descriptions of all active concepts that are active in the
-- Cardiovascular finding reference set
selectc.idasconceptid, d.idasdescriptionid, d.termas term
fromconcepts c, concept_refsetcr, descriptions d, description_refsetdr
where c.id = cr.referencedconceptid
and c.id = d.conceptid
and d.id = dr.referenceddescriptionid
and dr.valueid = 900000000000548007  -- ID of "preferred"
and cr.refsetid =32570381000036103-- ID of "Cardiovascular finding reference set"
and c.active = 1
and d.active = 1
and cr.active = 1
and dr.active = 1
order byd.term
```

conceptid	descriptionid	term
241154007	361157010	Abdominal angina
248713000	371226017	Abdominal aorta finding
233985008	350580017	Abdominal aortic aneurysm
75878002	126029017	Abdominal aortic aneurysm without rupture
233955003	350527016	Abdominal aortic atherosclerosis

Figure 6: Extract of a query – example

The interesting thing to note in this Query example is that it joins the Cardiovascular reference set, ADRS, concepts and descriptions table to achieve this, which is the minimum necessary as the active flag needs to be checked on each.

Also of interest is that the query assumes that the data loaded is a snapshot, and makes no reference to the effective time filed on any of the tables. This works because a snapshot only has one version for every entity – the latest one.

If this query was run against tables containing a Full release it would be necessary to add additional clauses to filter on the effective time.

4 Getting started

There are two distinct packages, one that runs on Java platforms, and the other that runs on .NET platforms. The following sections provide information on installing and using both of these applications.

4.1 System requirements – Java

As the application is a Java based, it will work on any platform that supports version 1.6.0 of Sun Microsystems Java^{TM} Development Kit (JDK).

The following items are required to build and run the sample application:

1 Sun Microsystems Java Development Kit (JDK) version 1.6.0 installed on the target machine. (Go to http://www.oracle.com/technetwork/java/javase/downloads/).

2 Apache Ant version 1.8 installed. (Go to http://ant.apache.org/).

Warning: This application will not run on Windows using Ant version 1.8.1 or later. It is a known issue that affects these versions of Ant. Please use a version such as Ant 1.8 or earlier if using Windows. Note that Linux and Apple systems may use the later versions.

3 Access to a MySQL Server 5.0 – 5.6 database. If not available, MySQL may need to be installed and configured. (Go to http://www.mysql.com/).

4 The SNOMED CT-AU Terminology Bundle.

4.2 Setup and configuration – Java

1 Before you run the Refset sample application you must first modify the *application.properties* file and specify:

a The name of the database server.

b The database schema name.

c The User Name to access the database.

d The Password to access the database.

e The fully qualified location/path of the extracted SNOMED CT-AU Terminology Bundle.

Warning: When running on the Windows operating system, the path separator in ANT is the forward double slash “\\”, for example: *rf2.source.dir =Z:\\Files\\NEHTA6112010_SNOMED_CT-AU_TerminologyReleaseFileBundle_20100531*

2 To run the ant build, open a command-line console and in the same directory as the build.xml file run “ant <*target*>”, where the target is one of the following:

Table 2: Ant commands

Command	Description
ant (no target)	Display this list of build targets.
ant db	Builds the database (creates the tables and loads in the SNOMED CT-AU RF2 files).
ant app	Compiles the application and run it. Use this target to avoid rebuilding the database every time.
ant test	Run the provided JUnit test cases to ensure the application is functioning correctly.

The system allows you to specify the amount of memory that is to be allocated for the application using the "Max Memory" option. If you are performing large requests, for example, a request that would return over 10 000 results, you may need to increase the size of the available memory.

4.3 System requirements – .NET

This application is designed to run on .NET 3.5 using Microsoft® Visual Studio® 2008 or 2010 Professional. It may also be run on Visual Studio 2008 and 2010 Express, although the Unit tests may not work.

MySQL Server isn't natively supported by Visual Studio, so you will also have to install a "connector" for the two products.

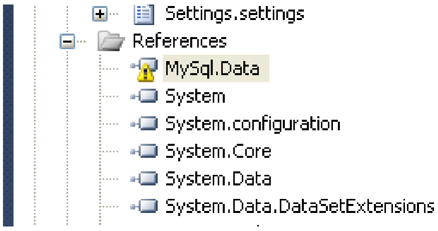
The following items are required to build and run the sample application:

1 Microsoft Visual Studio 2008 or 2010 Professional installed and running on the target machine. (Go to: http://www.microsoft.com/visualstudio/en-au/visual-studio-2010-launch/default.mspx/?qstr=CR_SCC=100343627&WT.srch=1).

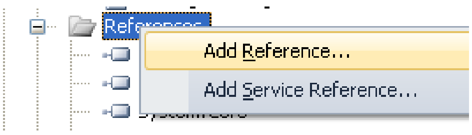
2 A minimum of MySQL Connect 6.3.5 installed and operational on the target machine. (Go to: http://dev.mysql.com/downloads/connector/net/).

Note: Depending on the version installed, the MySql.Data reference in Visual Studio may require updating. To update the system reference:

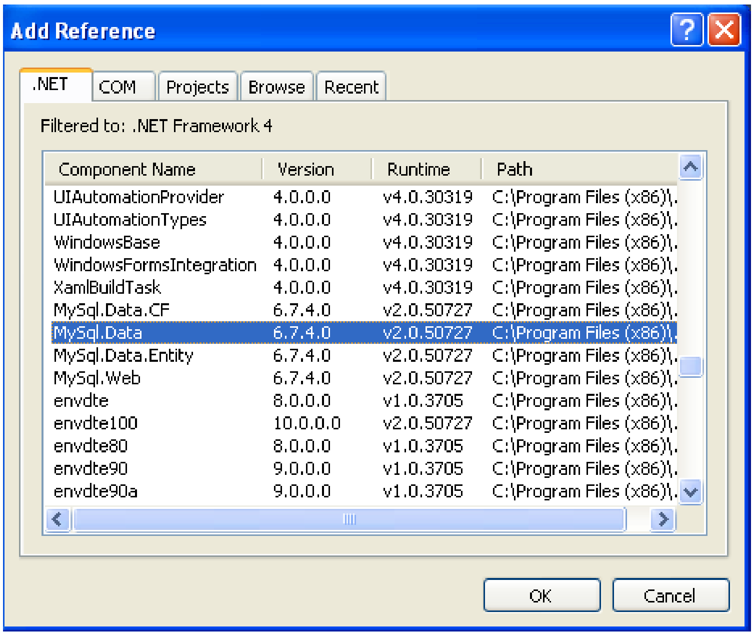
i. Remove existing reference from both CTDemo and LoadDB projects



ii. Re-add the reference to both projects by right clicking on References and selecting Add Reference



iii. Select MySql.Data under the .NET tab



Filtered to: .NET Framework 4

Component Name	Version	Runtime	Path
UIAutomationProvider	4.0.0.0	v4.0.30319	C:\Program Files (x86)\.
UIAutomationTypes	4.0.0.0	v4.0.30319	C:\Program Files (x86)\.
WindowsBase	4.0.0.0	v4.0.30319	C:\Program Files (x86)\.
WindowsFormsIntegration	4.0.0.0	v4.0.30319	C:\Program Files (x86)\.
XamlBuildTask	4.0.0.0	v4.0.30319	C:\Program Files (x86)\.
MySql.Data.CF	6.7.4.0	v2.0.50727	C:\Program Files (x86)\.
MySql.Data	6.7.4.0	v2.0.50727	C:\Program Files (x86)\.
MySql.Data.Entity	6.7.4.0	v2.0.50727	C:\Program Files (x86)\.
MySql.Web	6.7.4.0	v2.0.50727	C:\Program Files (x86)\.
envdte	8.0.0.0	v1.0.3705	C:\Program Files (x86)\.
envdte100	10.0.0.0	v2.0.50727	C:\Program Files (x86)\.
envdte80	8.0.0.0	v1.0.3705	C:\Program Files (x86)\.
envdte90	9.0.0.0	v1.0.3705	C:\Program Files (x86)\.
envdte90a	9.0.0.0	v1.0.3705	C:\Program Files (x86)\.

3 MySQL Server (Windows version) installed on the target machine. (Go to: http://dev.mysql.com/downloads/).

4 The SNOMED CT-AU Terminology Bundle.

4.4 Setup and configuration – .NET

This application is setup and configured using Visual Studio. There are two main projects that need to be configured:

1 LoadDB

2 CTDemo

4.4.1 LoadDB

LoadDB creates and builds the database (creates the tables and loads the SNOMED CT-AU RF2 files). It is the default start up project and contains the following editable properties:

- ConnectionString – the Server connection string.
- DatabaseName – the name of the database to connect to.
- SnomedFolderLocation – the location of the SNOMED CT folder.

4.4.2 CTDemo

CTDemo compiles the application and runs it. It contains the following editable properties:

- MaxRows – Limits the number of rows that are returned. It also affects the number of items returned when the “Find members by refset” option.
- ConnectionString – The database connection string.

To install and run the .NET version of the sample application:

1 Start Visual Studio.

2 Access the Solution Explorer, right click on the LoadDB project and select Properties.

3 Edit the properties to meet your system requirements.

4 Run the LoadDB project.

5 Right click on the CTDemo file in the Solution Explorer pane and select properties.

6 When you have saved the new settings, right click on the CTDemo file again and select “Set as StartUp Project” from the menu.

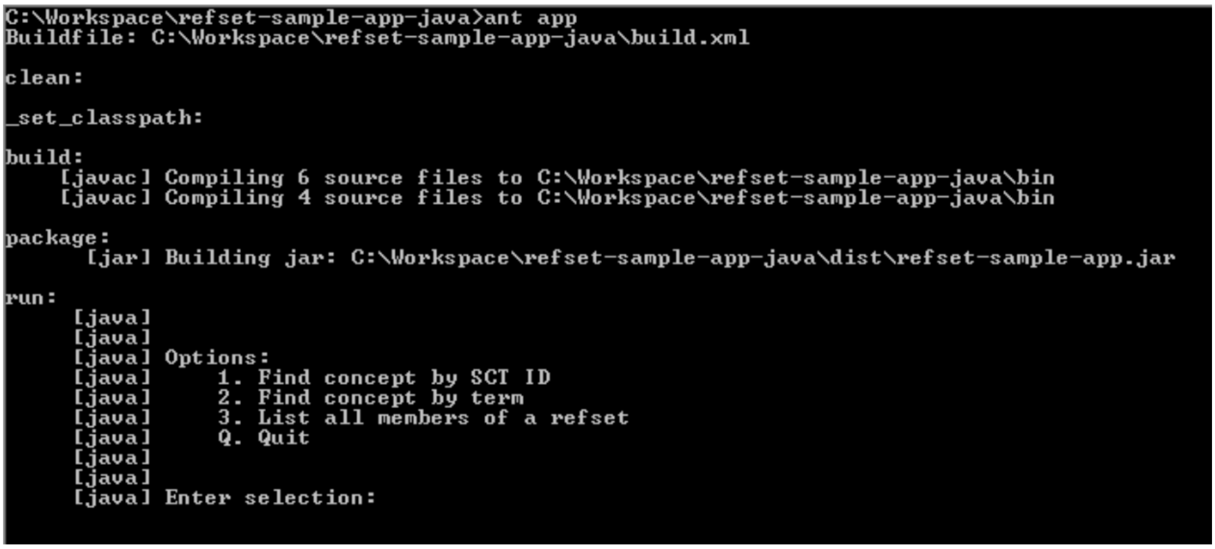
7 Edit the properties to meet your system requirements.

8 Ensure the application is working correctly by running the Unit Tests. To do this, select Test > Run > All Tests in Solution.

Note: You can also edit the properties of the two files by opening the app.config file.

5 Using the sample application

The following screenshot shows the main menu options available in the sample application:



```
C:\Workspace\refset-sample-app-java>ant app
Buildfile: C:\Workspace\refset-sample-app-java\build.xml

clean:

_set_classpath:

build:
    [javac] Compiling 6 source files to C:\Workspace\refset-sample-app-java\bin
    [javac] Compiling 4 source files to C:\Workspace\refset-sample-app-java\bin

package:
      [jar] Building jar: C:\Workspace\refset-sample-app-java\dist\refset-sample-app.jar

run:
     [java]
     [java]
     [java] Options:
     [java]     1. Find concept by SCT ID
     [java]     2. Find concept by term
     [java]     3. List all members of a refset
     [java]     Q. Quit
     [java]
     [java]
     [java] Enter selection:
```

Figure 7: Sample application options and command line

The interface of the Java version of the sample application and the .NET version appear similar; the only difference is the options are preceded by the text [java] in the java version. This is the only difference, and both versions operate in exactly the same way. To select an option, enter the corresponding number and press the **Enter** key.

There are three functions available in the Refset sample application:

1 Find concept by SNOMED CT identifier.

2 Find concept by term.

3 List all members of a Refset.

Note: Use option 2 first to find SNOMED CT-AU identifiers required for the other search options. You can search for the term “reference set” to find valid SNOMED CT identifiers (for use with option three of the sample application.

5.1 Find a Concept by a Term

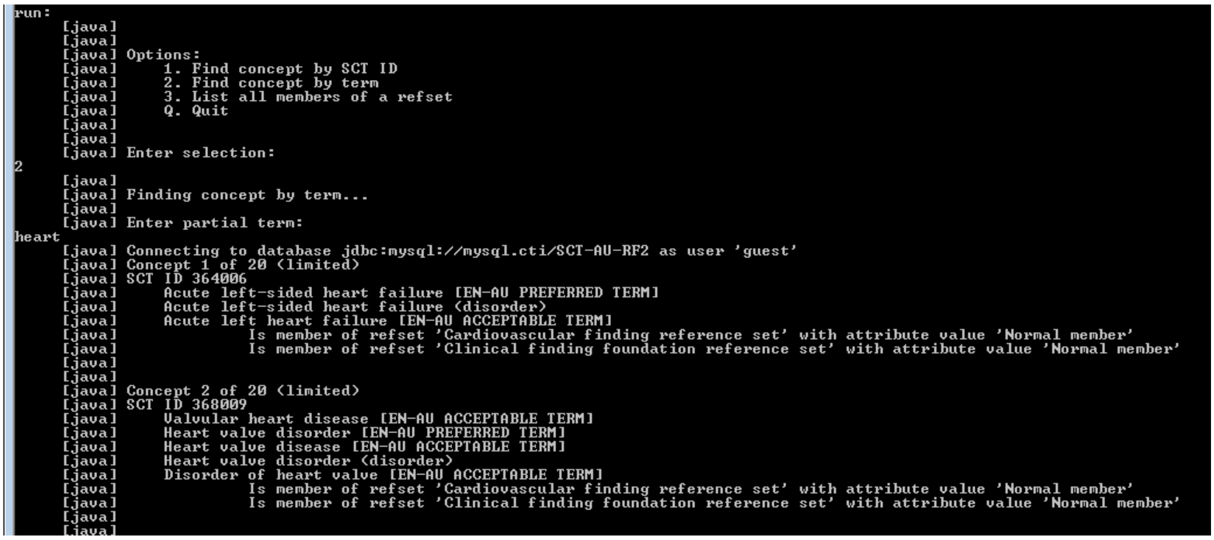
To find a concept by a term:

1 Type “2” and press **Enter**. The system prompts you to enter a term or fragment of a term that you wish to search on.

2 Type the term or fragment and press **Enter**.

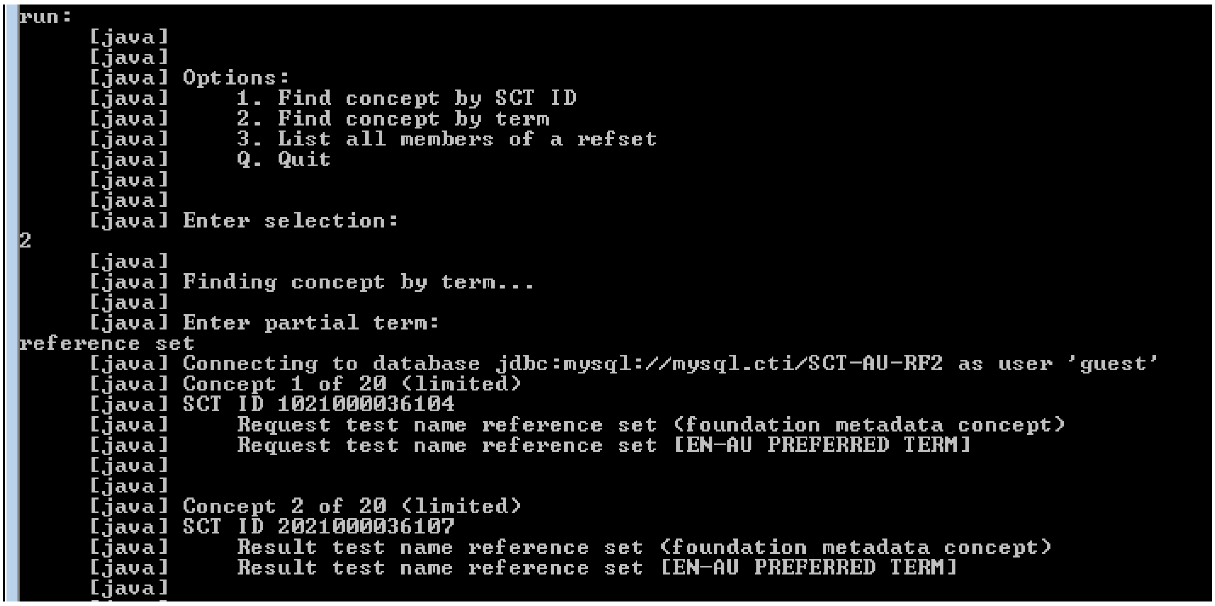
3 The system will list all concepts containing an active description matching the search term. For each concept it will display the concept’s SNOMED CT identifier and all of its active descriptions. It will indicate which descriptions are *preferred* or *acceptable* in the Australian Dialect. It will also list all the reference sets each concept is an active member of.

The number of search results that are display will be limited to the value of the "db.max.rows" entry specified in the application.properties configuration file. By default this is set to 20 concepts.



```
run:
     [java]
     [java]
     [java] Options:
     [java]     1. Find concept by SCT ID
     [java]     2. Find concept by term
     [java]     3. List all members of a refset
     [java]     Q. Quit
     [java]
     [java]
     [java] Enter selection:
2
     [java]
     [java] Finding concept by term...
     [java]
     [java] Enter partial term:
heart
     [java] Connecting to database jdbc:mysql://mysql.cti/SCT-AU-RF2 as user 'guest'
     [java] Concept 1 of 20 (limited)
     [java] SCT ID 364006
     [java]     Acute left-sided heart failure [EN-AU PREFERRED TERM]
     [java]     Acute left-sided heart failure (disorder)
     [java]     Acute left heart failure [EN-AU ACCEPTABLE TERM]
     [java]             Is member of refset 'Cardiovascular finding reference set' with attribute value 'Normal member'
     [java]             Is member of refset 'Clinical finding foundation reference set' with attribute value 'Normal member'
     [java]
     [java]
     [java] Concept 2 of 20 (limited)
     [java] SCT ID 368009
     [java]     Valvular heart disease [EN-AU ACCEPTABLE TERM]
     [java]     Heart valve disorder [EN-AU PREFERRED TERM]
     [java]     Heart valve disease [EN-AU ACCEPTABLE TERM]
     [java]     Heart valve disorder (disorder)
     [java]     Disorder of heart valve [EN-AU ACCEPTABLE TERM]
     [java]             Is member of refset 'Cardiovascular finding reference set' with attribute value 'Normal member'
     [java]             Is member of refset 'Clinical finding foundation reference set' with attribute value 'Normal member'
     [java]
     [java]
```

Figure 8: Example of a search for concepts related to the term "heart"



```
run:
     [java]
     [java]
     [java] Options:
     [java]     1. Find concept by SCT ID
     [java]     2. Find concept by term
     [java]     3. List all members of a refset
     [java]     Q. Quit
     [java]
     [java]
     [java] Enter selection:
2
     [java]
     [java] Finding concept by term...
     [java]
     [java] Enter partial term:
reference set
     [java] Connecting to database jdbc:mysql://mysql.cti/SCT-AU-RF2 as user 'guest'
     [java] Concept 1 of 20 (limited)
     [java] SCT ID 1021000036104
     [java]     Request test name reference set (foundation metadata concept)
     [java]     Request test name reference set [EN-AU PREFERRED TERM]
     [java]
     [java]
     [java] Concept 2 of 20 (limited)
     [java] SCT ID 2021000036107
     [java]     Result test name reference set (foundation metadata concept)
     [java]     Result test name reference set [EN-AU PREFERRED TERM]
     [java]
     [java]
```

Figure 9: Example of a search for concepts related to the term "reference set"

5.2 Find a Concept by SNOMED CT identifier

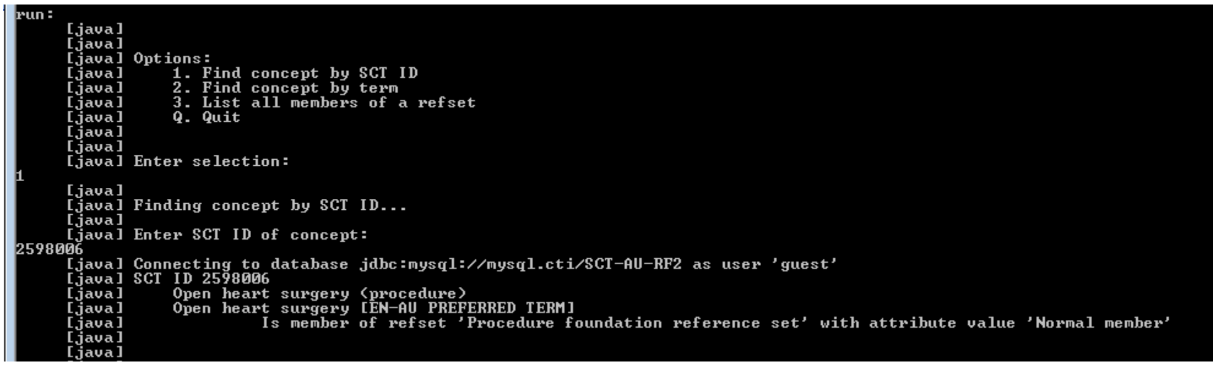
To find a concept using its SNOMED CT identifier

1 Type "1" and press **Enter**. The system prompts you to enter the SNOMED CT identifier of the concept you want to find.

2 Type the SNOMED CT identifier and press **Enter**.

3 The system will display the concept's SNOMED CT identifier and all of its active descriptions. It will indicate which descriptions are *preferred* or *acceptable* in

the Australian Dialect. It will also list all the reference sets the concept is an active member of.



```
run:
      [java]
      [java]
      [java] Options:
      [java]      1. Find concept by SCT ID
      [java]      2. Find concept by term
      [java]      3. List all members of a refset
      [java]      Q. Quit
      [java]
      [java]
      [java] Enter selection:
1
      [java]
      [java] Finding concept by SCT ID...
      [java]
      [java] Enter SCT ID of concept:
2598006
      [java] Connecting to database jdbc:mysql://mysql.cti/SCT-AU-RF2 as user 'guest'
      [java] SCT ID 2598006
      [java]      Open heart surgery (procedure)
      [java]      Open heart surgery [EN-AU PREFERRED TERM]
      [java]                Is member of refset 'Procedure foundation reference set' with attribute value 'Normal member'
      [java]
      [java]
```

Figure 10: Example of a search for concepts using a SNOMED CT identifier

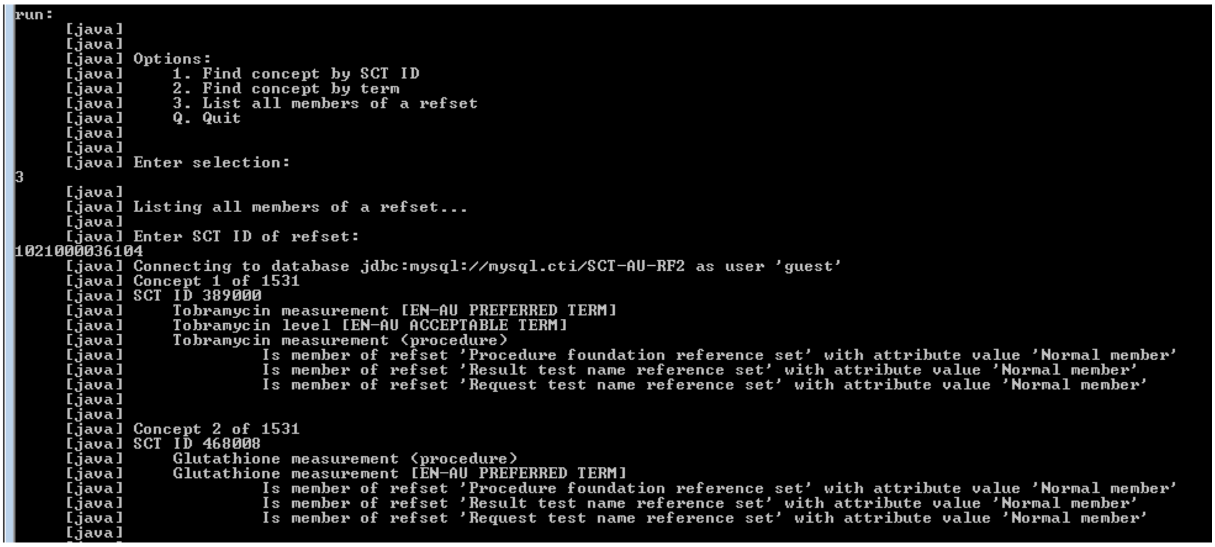
5.3 List all members of a reference set

To find a concept using its SNOMED CT identifier:

1 Type “3” and press **Enter**. The system prompts you to for the SNOMED CT identifier of the concept you want to find.

2 Type the SNOMED CT identifier of a reference set concept and press **Enter**.

3 The system will list all concepts that are active members of the reference set.



```
run:
      [java]
      [java]
      [java] Options:
      [java]      1. Find concept by SCT ID
      [java]      2. Find concept by term
      [java]      3. List all members of a refset
      [java]      Q. Quit
      [java]
      [java]
      [java] Enter selection:
3
      [java]
      [java] Listing all members of a refset...
      [java]
      [java] Enter SCT ID of refset:
1021000036104
      [java] Connecting to database jdbc:mysql://mysql.cti/SCT-AU-RF2 as user 'guest'
      [java] Concept 1 of 1531
      [java] SCT ID 389000
      [java]      Tobramycin measurement [EN-AU PREFERRED TERM]
      [java]      Tobramycin level [EN-AU ACCEPTABLE TERM]
      [java]      Tobramycin measurement (procedure)
      [java]                Is member of refset 'Procedure foundation reference set' with attribute value 'Normal member'
      [java]                Is member of refset 'Result test name reference set' with attribute value 'Normal member'
      [java]                Is member of refset 'Request test name reference set' with attribute value 'Normal member'
      [java]
      [java]
      [java] Concept 2 of 1531
      [java] SCT ID 468008
      [java]      Glutathione measurement (procedure)
      [java]      Glutathione measurement [EN-AU PREFERRED TERM]
      [java]                Is member of refset 'Procedure foundation reference set' with attribute value 'Normal member'
      [java]                Is member of refset 'Result test name reference set' with attribute value 'Normal member'
      [java]                Is member of refset 'Request test name reference set' with attribute value 'Normal member'
      [java]
```

Figure 11: Example of listing all members of a reference set

Appendix A Other resources

The following links may be useful if you want to learn more about implementing SNOMED CT:

- http://www.ihtsdo.org/publications/tools-for-terminology-developmentmaintenance/
- http://www.ihtsdo.org/publications/implementing-snomed-ct/implementation-guidance/
- https://csfe.aceworkspace.net (IHTSDO forums)

Glossary and Acronyms

Acronym	Term	Description
ADRS	**Australian Dialect Reference Set**	
	Delta release	This RF2 release type contains only the new and changed components between one release and the next.
	Full release	This RF2 release type contains every version of every component (concept, description, relationship, reference set member or identifier) ever published.
IHTSDO	**International Health Terminology Standards Development Organisation**	
RF1	**Release Format 1**	RF1 files contain the history of changes made to content within the terminology by never removing content from a release once it has been published. However content may be logically deleted by changing its status. Due to be discontinued in the July 2011 release from IHTSDO.
RF2	**Release Format 2**	RF2 contains the history of content changes over time. It has three release types, Full, Snapshot and Delta.
	Snapshot release	Similar to RF1, this RF2 release type contains only the most recent state of every component ever published.
SNOMED CT-AU	**SNOMED CT Australian Release**	SNOMED CT-AU is the Australian extension to SNOMED CT; the integrated national release of SNOMED CT for implementation in Australian deployed clinical IT systems.
SNOMED CT	**Systematized Nomenclature Of Medicine, Clinical Terms**	The clinical terminology maintained and distributed by the IHTSDO, created as a result of the merger of SNOMED RT and Clinical Terms Version 3[5].
TIG	**Technical Implementation Guide**	
TRG	**Technical Reference Guide**	
UUID	**Universal Unique Identifier**	

[5] Definition taken from the glossary provided on the IHTSDO website.