Digitally signed by National E-Health Transition Authority Ltd
Date: 2016.05.24 14:34:50 +10'00'

National Clinical Terminology Information Service

Australian Technical Implementation Guide v1.0

23 May 2016

Approved for external information

Document ID: NEHTA-2307:2016

National E-Health Transition Authority

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

Acknowledgements

Council of Australian Governments
The National E-Health Transition Authority is jointly funded by the Australian Government and all State and Territory Governments.

IHTSDO (SNOMED CT)
This material includes SNOMED Clinical Terms™ (SNOMED CT®) which is used by permission of the International Health Terminology Standards Development Organisation (IHTSDO). All rights reserved. SNOMED CT® was originally created by The College of American Pathologists. "SNOMED" and "SNOMED CT" are registered trademarks of the IHTSDO, (http://www.ihtsdo.org/).

HL7 International
This document includes excerpts of HL7™ International standards and other HL7 International material. HL7 International is the publisher and holder of copyright in the excerpts. The publication, reproduction and use of such excerpts is governed by the HL7 IP Policy (see http://www.hl7.org/legal/ippolicy.cfm) and the HL7 International License Agreement. HL7 and CDA are trademarks of Health Level Seven International and are registered with the United States Patent and Trademark Office. FHIR is a registered trademark of Health Level Seven International.

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	Head of National Services Operation and Management
Contact for enquiries	NEHTA Help Centre t: 1300 901 001 e: help@nehta.gov.au

Table of contents

1 Introduction

1.1 Purpose

This document provides practical implementation guidance for technical consumers (such as software developers and the general community of practice) on the use of SNOMED CT-AU and the Australian Medicines Terminology (AMT). This guide aims to provide:

- a brief introduction to SNOMED CT-AU and the AMT;
- guidance on using and interpreting the terminologies; and
- sample methods of querying the terminology content to retrieve data relevant to common use cases.

1.2 Intended audience

This document should be read by:

- vendors of technical healthcare products or systems; and
- software developers or testers who are responsible for producing, assuring or maintaining products that integrate with SNOMED CT-AU and the AMT.

The reader is assumed to have a basic understanding of software development, database management, and SNOMED CT.

1.3 Scope

A substantial amount of documentation already exists for SNOMED CT, SNOMED CT-AU, and the AMT, including implementation guidance. This document is designed to avoid overlapping existing documentation insofar as is feasible.

1.3.1 SNOMED CT

Implementation advice for SNOMED CT is provided by the IHTSDO's *SNOMED CT Technical Implementation Guide* [1] (hereafter: *SNOMED TIG*). Australian developers should use this document and the *SNOMED TIG* together. Cross references to specific sections within the *SNOMED TIG* are provided at various points in this document.

1.3.2 SNOMED CT-AU

This document supersedes the *SNOMED CT-AU Australian Implementation Guidance v2.0* [2]. It does not focus on the development of release artefacts or products. For example, the *NCTIS Reference Set Library* [3] and *NCTIS Development Approach for Reference Sets* [4] describe products that are part of the release whereas this document is intended for guidance on the options to implement certain aspects of SNOMED CT-AU.

1.3.3 AMT

This version of this document partially supersedes the *AMT Technical Implementation Guide* [5] (hereafter: *AMT TIG*). However, the *AMT TIG* includes some material that has not been incorporated into this document. This other material will eventually be incorporated into other documents and the *AMT TIG* will be deprecated.

Therefore, the *AMT TIG* can be consulted alongside this document, with appropriate caution. If there are any discrepancies between the two, this document should be regarded as the source of truth. Contact NEHTA at help@nehta.gov.au if you require clarification on any related issues.

This document does not cover a business overview of the AMT product or drivers for implementing the AMT product in general. These topics are covered by the *AMT v3 Overview and Use Cases* [6].

1.4 Clinical terminology overview

Clinical terminology contributes to the improvement of healthcare through supporting the recording, display and exchange of healthcare information and the ability to deliver decision support services to healthcare providers. Healthcare consumers benefit from the use of terminology to more clearly describe and accurately record their healthcare information. The application of clinical terminology has a range of benefits, including:

- Clinical efficiency and a consistent vocabulary across all healthcare domains.
- Reduced error rates and better recording of clinical information at the required level of granularity.
- Consistent retrieval, exchange and analysis of recorded clinical information.
- Reduced risk of incorrect interpretation of clinical information.

In addition, clinical terminology supports or enables:

- Semantic interoperability between disparate clinical information systems.
- Reusability of clinical information (e.g. record once, use many times).
- Consistent representation of clinical terms.
- Machine processing of clinical information.
- Extensibility, which in turn enables the terminology to improve and evolve to meet changing needs.

These benefits are major drivers for organisations to adopt terminology. However, to support the realisation of these benefits, those working to develop, integrate and maintain terminology within a healthcare software system require a comprehensive understanding of the ontology. This is not insignificant given the amount and, at times, complex nature of the information that needs to be understood. Areas of coverage include, but are not limited to, file formats, terminology components, relationship types, hierarchies, reference sets and the interaction between the terminology and the information model.

Terminology adoption requires much more than just an in-depth understanding of terminology. Various groups of skilled professionals from different backgrounds and knowledge domains are needed to support the adoption process.

1.5 SQL examples

Where possible, example SQL scripts are provided to assist in the demonstration of how features work. The schema for these code samples is configured using the **Australian_Terminology_Sample_Scripts.zip** file.

The accompanying query examples herein use a Snapshot release and focus on active content only, unless otherwise indicated.

Please note that all code samples and sample scripts provided in this document are for demonstration purposes only and may not represent the most efficient or robust implementation approach. Implementers are advised to conduct their own performance tuning and ensure appropriate exception handling.

2 Overview of Australian terminologies

2.1 SNOMED CT-AU overview

SNOMED CT Australian Release (SNOMED CT-AU) is the Australian extension to SNOMED CT, providing local variations and customisations of terms relevant to the Australian healthcare community. It includes core content from the international release along with additional content developed to meet the requirements for use within Australian clinical IT systems.

Any new content from the international release is reviewed to ensure it is suitable and relevant for local use in Australia. Periodic reviews on existing data are also conducted to improve the quality and suitability for Australian use.

SNOMED CT-AU also contains locally created content, which may be requested by external and internal stakeholders or when required to support implementations. New content can include new concepts and terms, reference sets and documents to provide guidance and support for implementations. Content is often created collaboratively as small projects with external stakeholders including jurisdictions, vendors and clinicians.

The SNOMED CT-AU release bundle contains the relevant SNOMED CT International components as well as the content developed by NEHTA for use in Australian health care settings. This entails the provision of Australian Preferred Terms and other Australian-specific content, which are modelled consistently in line with SNOMED CT principles. Access to SNOMED CT-AU is provided without charge by NEHTA to Australian licence holders.

2.2 AMT overview

The Australian Medicines Terminology (AMT) is the national terminology that delivers unique codes to unambiguously identify originator and generic brands of medicines commonly used in Australia. It also provides standard naming conventions and terminology to accurately describe medications.

The key aim of the AMT is to provide a consistent and safe approach to the identification and naming of medicines, which can support medicines management and activity across the entire Australian health domain. The AMT may be used for documenting prescribe/dispense/administration actions or maintaining general medication records for transfer of information.

The AMT is an extension of SNOMED CT, created specifically to address the objectives of:

- providing a consistent identification of branded and generically equivalent medicines; and
- using consistent naming conventions and terminology to describe medicines and to facilitate searching for medicines in clinical information systems.

NEHTA continues to work with relevant stakeholders, as well as national and international clinicians and terminology experts, to further refine the specifications, editorial rules, standards and infrastructure necessary to achieve these aims and objectives.

New content for the AMT is released on a monthly basis and includes updates from the Pharmaceutical Benefits Scheme (PBS) and Therapeutic Goods Administration (TGA).

The AMT is published as part of SNOMED CT-AU, as well as (currently) being available as a smaller stand-alone bundle.

2.2.1 Scope of the AMT

The scope of the AMT is to include medicines that may be encountered in the Australian healthcare environment. The AMT includes identifiers and descriptions for:

- The majority of "Registered" (AUSTR) products contained in the Australian Register of Therapeutic Goods.
- Prioritised "Listed" medicines (AUSTL) by the TGA.
- Other medicines and therapeutic products required to support AMT use cases. The AMT also includes non-approved therapeutic goods, for example, medicines available under the Special Access Scheme.

At present the AMT does not contain all of the products in the categories listed above. The addition of new products is prioritised based on feedback from end users.[1]

The structure of the AMT is based upon seven "notable concepts", of which two can take tangible form: Trade Product Unit of Use (TPUU) and Containered Trade Product Pack (CTPP). The former is an individual dose form without any accompanying container information, such as a single tablet, and the latter adds container information, e.g. "bottle".

Packs are also provided without a specific container (Trade Product Pack - TPP) and brand (Medicinal Product Pack - MPP).

TPUUs have attributes defining their ingredients (including strength), form, and unit of use size. Medicinal Product Unit of Use (MPUU) are provided as the unbranded form of TPUU, and Medicinal Products (MP) represent specific combinations of ingredients (devoid of form, strength and unit of use size). Trade Products (TPs) represent the brand names of products.

The concepts are organised as a hierarchy such that concepts become more specific further deeper down the tree, as found across the whole of SNOMED CT-AU. Section 3 provides further details of the logical model and structure of the terminology.

2.2.2 Out of scope

There is a wide range of knowledge about medicines that is not included in the AMT. Some of this information may be provided by third party knowledge or decision support systems and can be linked to the AMT or SNOMED CT-AU terminology concepts.

[1] See the NCTIS request submission page at http://www.nehta.gov.au/our-work/clinical-terminology/request-submission-product-content-changes.

Examples of information drawn from knowledge bases that are not within the scope of the AMT include, but are not limited to:

- potential adverse effects;
- cautionary and advisory label recommendations and instructions; and
- interactions between drug and diseases, food or other drug excipients.

Please note that excipients will not be modelled in the AMT unless presented with a clear use case that is agreed to by the relevant NEHTA governance body or bodies.

A Medicinal Product will only define inactive (inert) ingredients where these are part of sequential multicomponent products, or diluents provided for the preparation of the actual administrable form of a product.

3 Concept models

3.1 SNOMED CT-AU concept model

SNOMED CT terminologies define concept models for different hierarchies of content which define patterns and rules for concepts in those hierarchies. Concept models define the structure and organisation of clinical terminologies; an analogy could be drawn with the way that database schemas define the structure relational data.

Each concept model essentially specifies the pattern of logic used to define concepts within that domain. For example, concept models define rules for a hierarchy or group of hierarchies that govern:

- the types of relationships used to define concepts;
- the combinations and groupings of relationships used to define concepts; and
- which types of concepts will be “defined” and which will be “primitive”.

SNOMED CT facilitates a hierarchical and polymorphic structure of medical nomenclature, defined as concepts. The concept hierarchy allows for concepts on different lineages to join where a particular concept can be described as any (or all) of two or more different concepts. (For example, *Laparoscopic cholecystectomy* is not only an endoscopic operation; it is also a cholecystectomy as well as a laparoscopic procedure.) This structure also provides greater specificity as it goes deeper, starting from the most basic or aggregate concepts at the top of the hierarchy to the most singular at the lower levels.

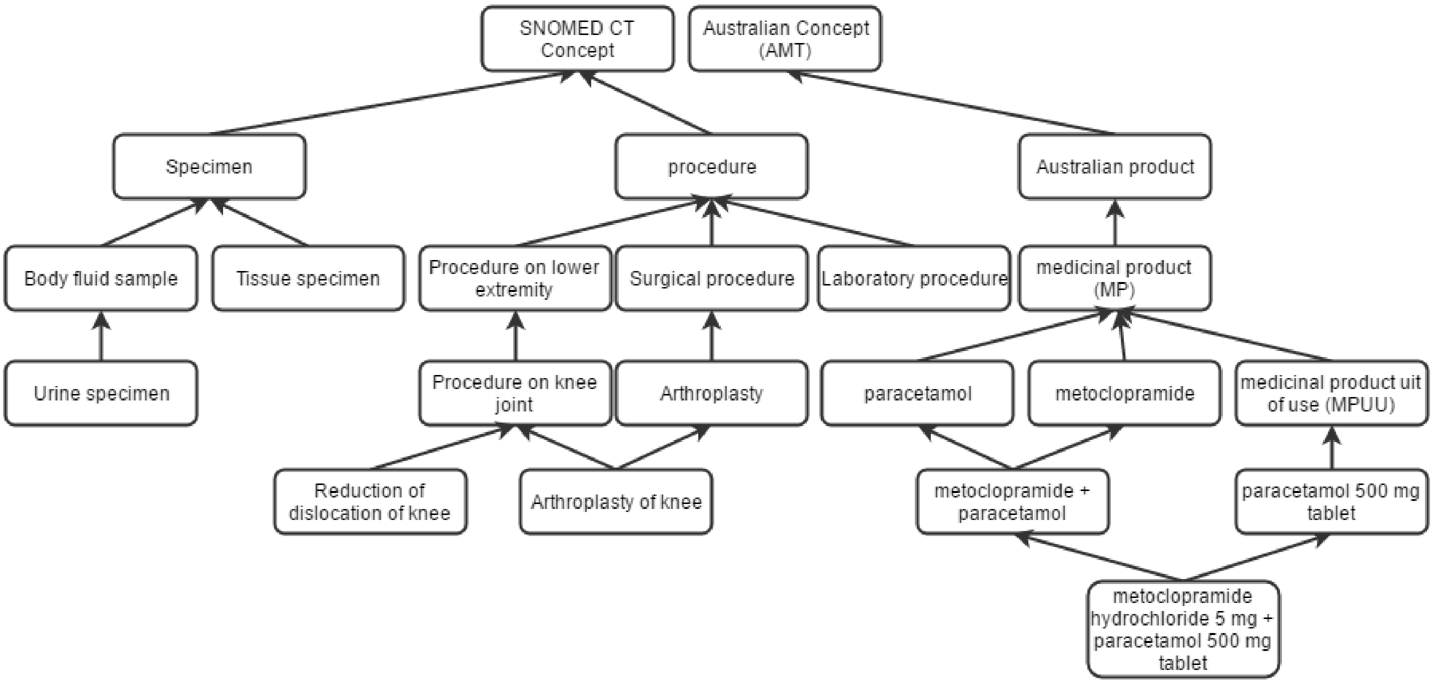


Figure 1: Example conceptual view of the SNOMED CT AU hierarchy (not all intermediate concepts are shown)

Fundamentally, the terminology is comprised of concepts, descriptions, and relationships, with the objective of precisely representing clinical information across the full range of health care settings.

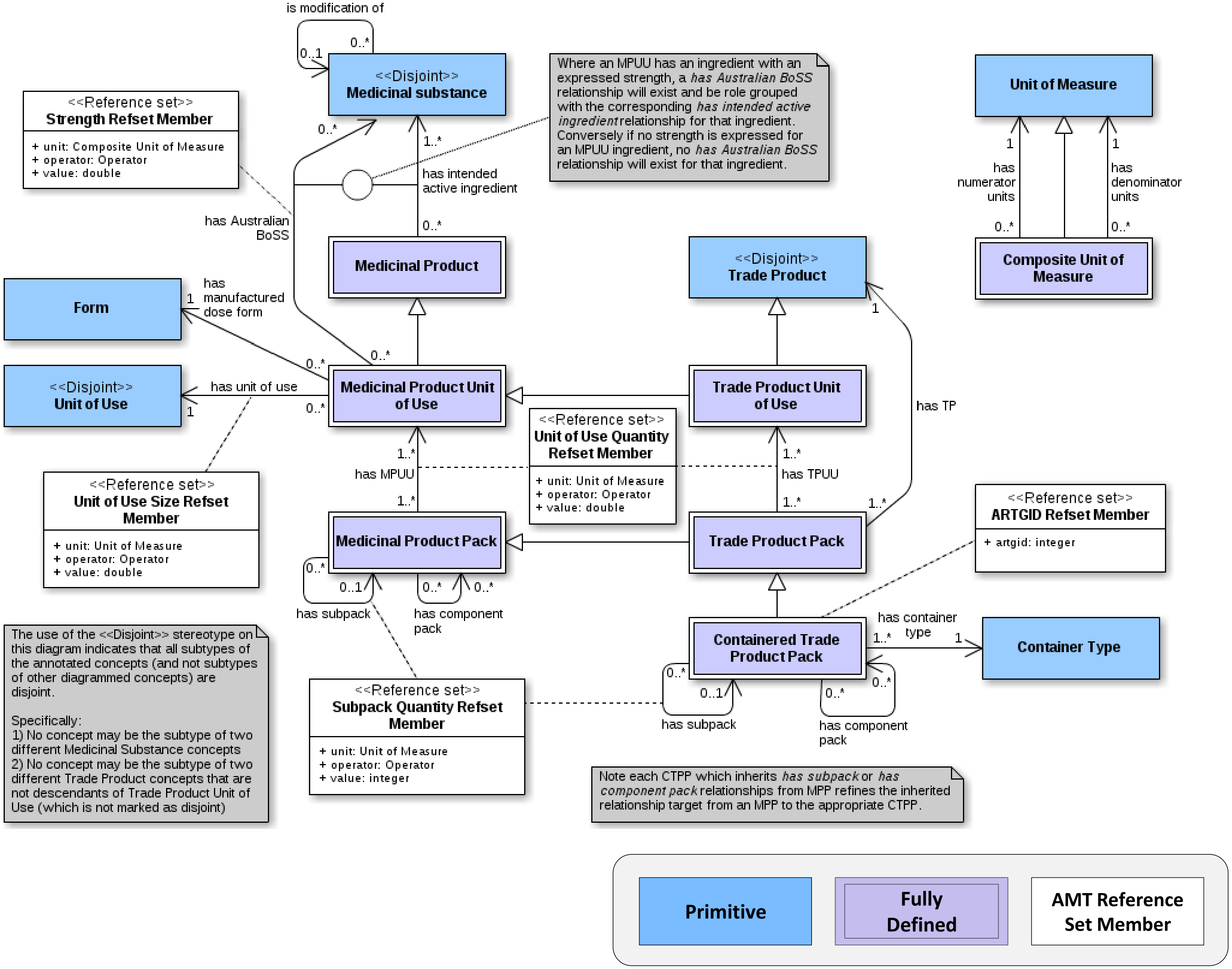
Table 1: Foundations of clinical terminologies

Component	Description
Concepts	Represents the clinical concepts that make up SNOMED CT. A concept is given meaning by its Fully Specified Name, sourced from the Description table. A concept may be distinguished from or refined by association with other concepts using relationships, which are held in the Relationships table.
Descriptions	Holds descriptions that describe each SNOMED CT concept. A description is used to give meaning to a concept and provide well-understood and standard ways of referring to a concept. As well as the Fully Specified Name, the Description table includes one or more Synonyms that can be on clinical interfaces.
Relationships	Provides information about the relationships between the concepts. These relationships define and bring meaning to the individual concepts relative to other concepts.

3.2 AMT concept model

Figure 2 below provides an illustrative summary of the AMT concept model. This section provides a brief highlight of parts of the AMT that are markedly different from the rest of SNOMED CT-AU. Further details concerning the AMT are available in other documents:

- Full details of the model are described in the *AMT Technical Implementation Guide* [5]
- The use cases that the AMT supports are described in the *AMT v3 Overview and Detailed Business Use Cases* [6].
- The naming conventions of AMT descriptions and full definitions of the AMT classes are described in the *AMT Editorial Rules* [7].



Where an MPUU has an ingredient with an expressed strength, a *has Australian BoSS* relationship will exist and be role grouped with the corresponding *has intended active ingredient* relationship for that ingredient. Conversely if no strength is expressed for an MPUU ingredient, no *has Australian BoSS* relationship will exist for that ingredient.

The use of the <<Disjoint>> stereotype on this diagram indicates that all subtypes of the annotated concepts (and not subtypes of other diagrammed concepts) are disjoint.

Specifically:
1) No concept may be the subtype of two different Medicinal Substance concepts
2) No concept may be the subtype of two different Trade Product concepts that are not descendants of Trade Product Unit of Use (which is not marked as disjoint)

Note each CTPP which inherits *has subpack* or *has component pack* relationships from MPP refines the inherited relationship target from an MPP to the appropriate CTPP.

Figure 2: The AMT concept model

3.2.1 Seven notable product concepts

The seven product classes also known as the "seven notable concepts" are central to the AMT model. These classes group concepts representing different abstractions of branded products and their generic product equivalents at various levels of granularity. Figure 3 includes a brief definition of the seven product classes along with an example of the concept for each class. Full definitions and more worked examples can be found in the *AMT Editorial Rules* [7].

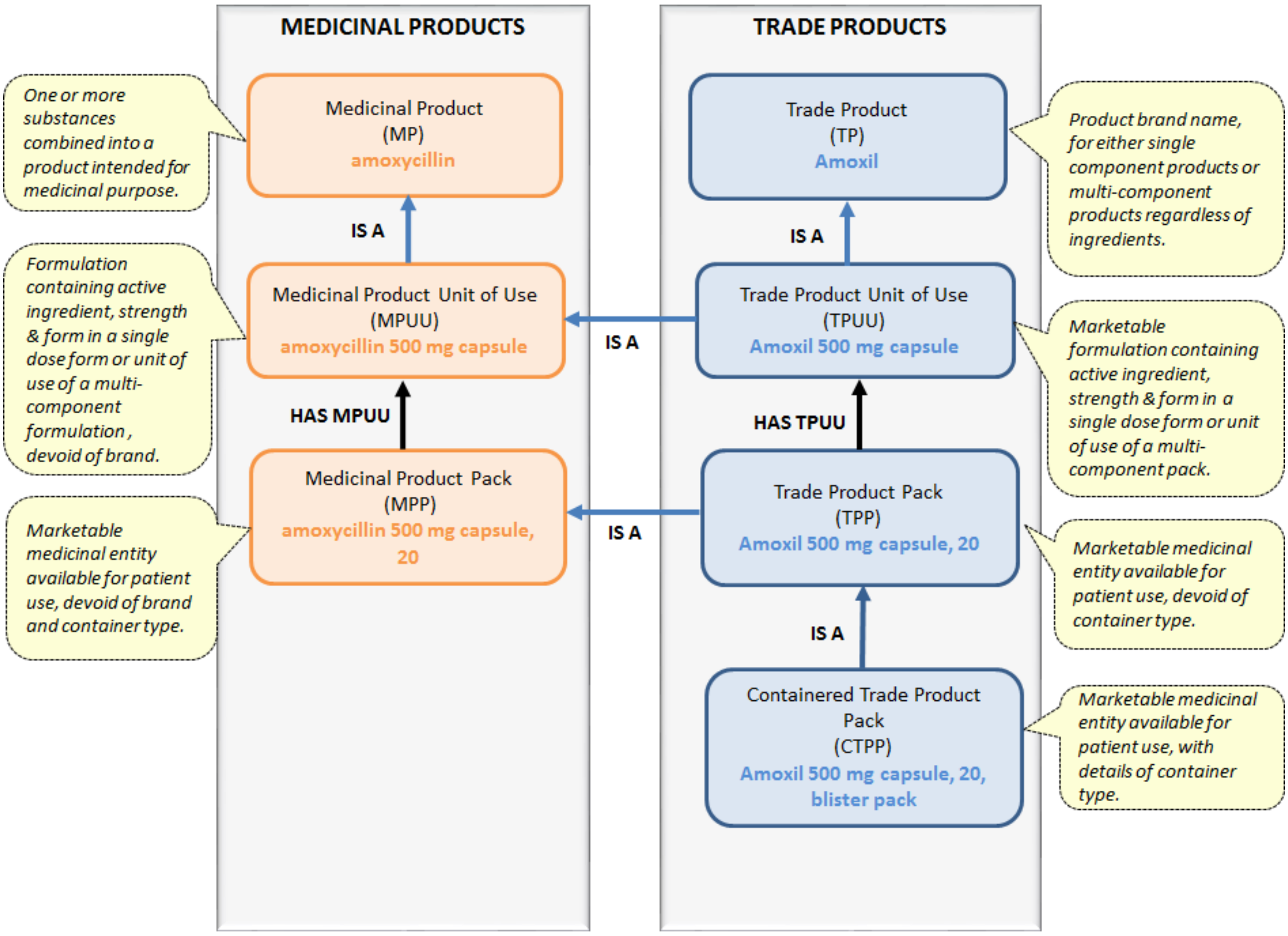

MEDICINAL PRODUCTS
TRADE PRODUCTS
One or more substances combined into a product intended for medicinal purpose.
Medicinal Product (MP) amoxycillin
Trade Product (TP) Amoxil
Product brand name, for either single component products or multi-component products regardless of ingredients.
IS A
IS A
Formulation containing active ingredient, strength & form in a single dose form or unit of use of a multi-component formulation, devoid of brand.
Medicinal Product Unit of Use (MPUU) amoxycillin 500 mg capsule
IS A
Trade Product Unit of Use (TPUU) Amoxil 500 mg capsule
Marketable formulation containing active ingredient, strength & form in a single dose form or unit of use of a multi-component pack.
HAS MPUU
HAS TPUU
Marketable medicinal entity available for patient use, devoid of brand and container type.
Medicinal Product Pack (MPP) amoxycillin 500 mg capsule, 20
IS A
Trade Product Pack (TPP) Amoxil 500 mg capsule, 20
Marketable medicinal entity available for patient use, devoid of container type.
IS A
Containered Trade Product Pack (CTPP) Amoxil 500 mg capsule, 20, blister pack
Marketable medicinal entity available for patient use, with details of container type.

Figure 3: Seven notable concepts of the AMT and their definitions

4 Format of the release bundle

4.1 Components of an RF2 release

The combined SNOMED CT-AU and AMT content is distributed in an end product file using the naming convention:

EP_<identifier>_<YYYY>_ClinicalTerminology_v<releaseDate>.zip

For example:

EP_2227_2016_ClinicalTerminology_v20160131.zip

This file contains the components listed in the table below.

Table 2: Contents of the terminology release bundle

Component	Naming convention	Example
Release note	**NEHTA_<identifier>_<YYYY>_ClinicalTerminology-SNOMEDCT-AU_ReleaseNote_v<ReleaseDate>.pdf**	**NEHTA_2223_2016_ClinicalTerminology-SNOMEDCT-AU_ReleaseNote_v20160131.pdf**
Combined release file with SNOMED CT-AU and AMT data	**NEHTA_<identifier>_<YYYY>_SNOMEDCT-AU_CombinedReleaseFile_v<ReleaseDate>.zip**	**NEHTA_2225_2016_SNOMEDCT-AU_CombinedReleaseFile_v20160131.zip**
Stand-alone release file for AMT data only	**NEHTA_<identifier>_<YYYY>_AustralianMedicinesTerminology_DataExtract_v<ReleaseDate>.zip**	**NEHTA_2224_2016_AustralianMedicinesTerminology_DataExtract_v20160131.zip**

The schematic diagram below illustrates the folder structure within the release file.

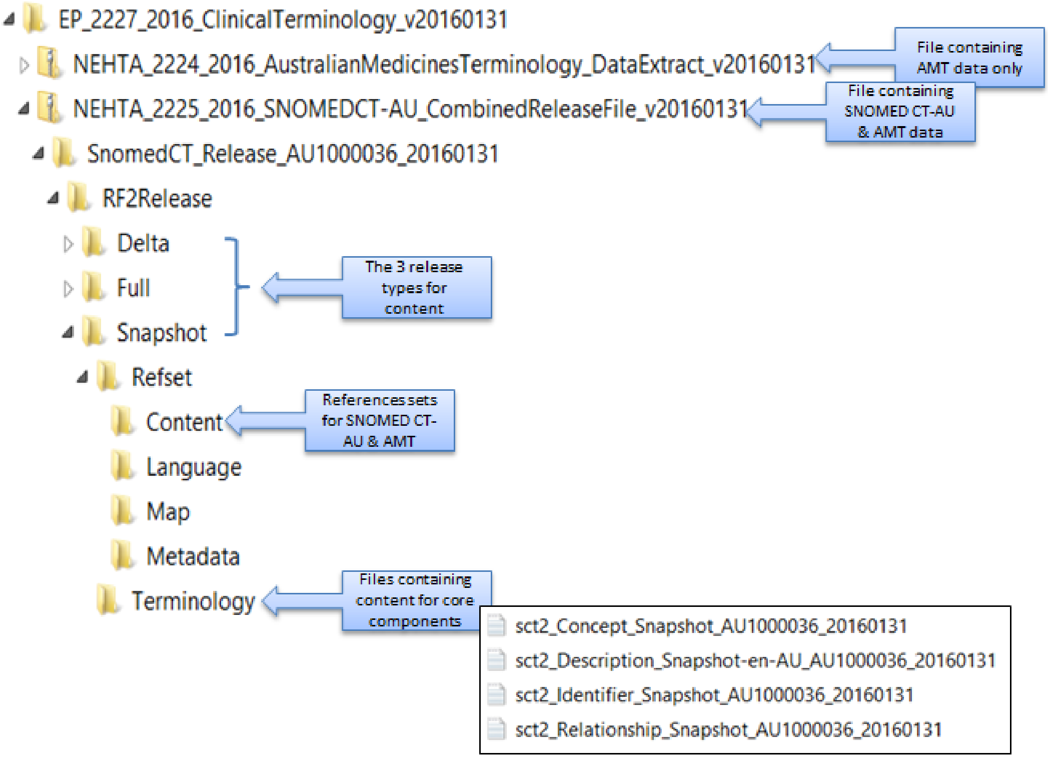

EP_2227_2016_ClinicalTerminology_v20160131
NEHTA_2224_2016_AustralianMedicinesTerminology_DataExtract_v20160131
File containing AMT data only
NEHTA_2225_2016_SNOMEDCT-AU_CombinedReleaseFile_v20160131
File containing SNOMED CT-AU & AMT data
SnomedCT_Release_AU1000036_20160131
RF2Release
Delta
Full
Snapshot
The 3 release types for content
Refset
Content
References sets for SNOMED CT-AU & AMT
Language
Map
Metadata
Terminology
Files containing content for core components
sct2_Concept_Snapshot_AU1000036_20160131
sct2_Description_Snapshot-en-AU_AU1000036_20160131
sct2_Identifier_Snapshot_AU1000036_20160131
sct2_Relationship_Snapshot_AU1000036_20160131

Figure 4: Directory layout of release bundle

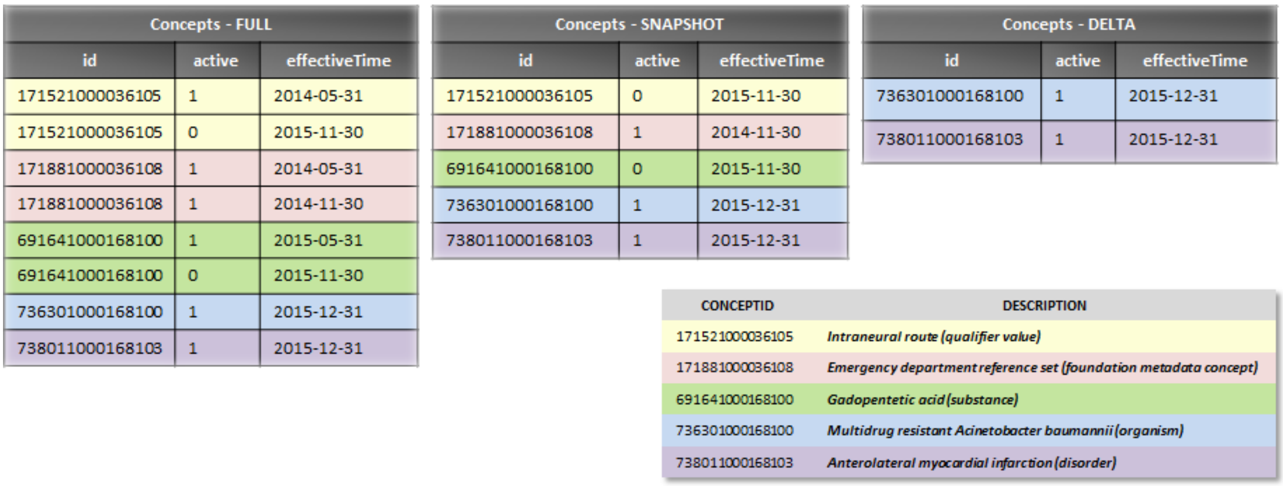
Comprehensive details of the RF2 specification are available in Section 5.3 "Release Format 2 – Introduction" of the *SNOMED TIG* [1].

4.2 RF2 distribution types

The RF2 distribution types is available in three variants providing different views of the terminologies history, as summarised in the following table and illustrated in Figure 5.

Table 3: Terminology release types

Release Type	Description
Full	The files representing each type of component contain every version of every component ever released.
Snapshot	The files representing each type of component contain one version of every component released up to the time of the snapshot. The version of each component contained in a snapshot is the most recent version of that component at the time of the snapshot.
Delta	The files representing each type of component contain only component versions created since the previous release. Each component version in a delta release represents either a new component or a change to an existing component.



Concepts - FULL

id	active	effectiveTime
171521000036105	1	2014-05-31
171521000036105	0	2015-11-30
171881000036108	1	2014-05-31
171881000036108	1	2014-11-30
691641000168100	1	2015-05-31
691641000168100	0	2015-11-30
736301000168100	1	2015-12-31
738011000168103	1	2015-12-31

Concepts - SNAPSHOT

id	active	effectiveTime
171521000036105	0	2015-11-30
171881000036108	1	2014-11-30
691641000168100	0	2015-11-30
736301000168100	1	2015-12-31
738011000168103	1	2015-12-31

Concepts - DELTA

id	active	effectiveTime
736301000168100	1	2015-12-31
738011000168103	1	2015-12-31

CONCEPTID	DESCRIPTION
171521000036105	*Intraneural route (qualifier value)*
171881000036108	*Emergency department reference set (foundation metadata concept)*
691641000168100	*Gadopentetic acid (substance)*
736301000168100	*Multidrug resistant Acinetobacter baumannii (organism)*
738011000168103	*Anterolateral myocardial infarction (disorder)*

Figure 5: Examples of Full, Snapshot and Delta content

These forms are useful in different contexts and maintenance strategies. For example:

- The Snapshot form is easiest to query as it contains just the latest version of each component. Data updates for a Snapshot implementation typically require replacing content with a new Snapshot release.
- Queries written against the Full form are more complex, having to filter out the historical versions of each component. However, data updates are simplified to appending the next Delta release to the existing Full data release.

Implementers need only use the forms applicable to their implementation approach. A typical implementation would start with either a Snapshot or an initial Full release followed by Delta updates. The *SNOMED TIG* provides further details concerning importing the various release types in Section 7.2 “Importing SNOMED CT release data”.

5 Importing terminology content

Before consuming terminology codes in a clinical software system, the content needs to be loaded into a database repository. The figure below outlines the general steps for the overall import process, starting with obtaining the release file, right through to creating the tables and then loading in the data.

The details of creating the main tables and other related objects are supplied in the **Australian_Terminology_Sample_Scripts.zip** file.[2] These objects need to be created prior to loading content into the respective tables.

The file contains configuration scripts for the set-up of a sample database schema into which the content of the Australian Terminology Release Bundle is loaded.

The purpose of these scripts is to serve as a learning tool to understand the release format and data structures of the terminologies within a relational database environment. They provide one approach for implementing SNOMED CT-AU and the AMT. However, other approaches are possible and each needs to be given due consideration for each specific implementation.

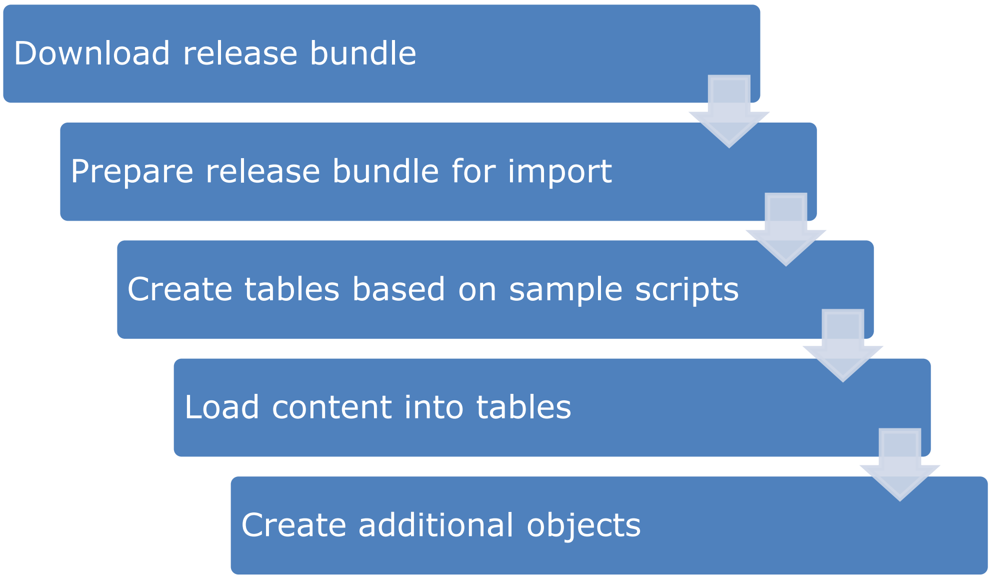

Download release bundle
Prepare release bundle for import
Create tables based on sample scripts
Load content into tables
Create additional objects

Figure 6: Process for loading terminology content

5.1 Download the release bundle

1. A licence is required to download and use SNOMED CT-AU. This can be easily obtained by registering a request via the Online Registration form on the NEHTA website at http://www.nehta.gov.au/our-work/clinical-terminology/registering-for-a-license
2. Having acquired the appropriate licences, the Australian Clinical Terminology release bundle can be downloaded from the NEHTA website at https://www.nehta.gov.au/implementation-resources/ehealth-foundations/clinical-terminology
3. Additionally the **NCTIS_Australian_Terminology_Sample_Scripts.zip** file needs to be downloaded, available at

[2] Available at: https://www.nehta.gov.au/implementation-resources/ehealth-foundations/snomed-ct-au-common

https://www.nehta.gov.au/implementation-resources/ehealth-foundations/snomed-ct-au-common

5.2 Prepare release bundle for import

1. Save the downloaded copy of the release bundle and sample scripts onto a server hosting the database repository for terminology content.
2. Uncompress the primary release bundle, which in turn will produce two more zip files.
3. Select either the combined release file (containing both SNOMED CT-AU and the AMT) or the AMT data extract file.
4. Uncompress the selected file.

It is advisable to read the release note contained within the main content bundle to find out what changes have been made in that release.

5.3 Create tables based on sample scripts

1. Connect to the database repository hosting the terminology content.
2. Execute the **1_createSchema.sql** script found in the **Australian_Terminology_Sample_Scripts.zip** file to set up a database and create common tables to store SNOMED CT-AU and AMT content.

5.4 Load content into the tables

The sample scripts contain a file to load data from the release bundle into the newly created tables. The **2_populateTables.sql** file needs to be modified to replace the general paths with the location of the downloaded terminology release file.

5.5 Create additional objects

The final step in configuring a database to store terminology content is to create additional objects to improve query performance. Additionally, there are several AMT-specific objects that may be created to support specific characteristics of its concept model.

3_createIndexes.sql	Creates indexes to improve the performance of SQL queries against the base tables.
4_createRoutines.sql	Creates functions that simplify and speed up access to the FSN and PT terms.
5_createTransitiveClosure.sql	Creates a transitive closure table to store all the IS A associations separate from the relationships file.

6_createAMTObjects.sql

Creates AMT-specific objects such as:

- Views to preselect the notable concepts.
- Primary table to store concrete domain reference sets.
- Views to preselect each of the concrete domains.
- Precoordinated tables to store the results of complex queries.

7_populateAMTObjects.sql

Loads content into the AMT-specific tables and views using either LOAD or INSERT INTO SELECT… commands.

8_createAMTIndexes.sql

Creates indexes on the AMT-specific tables.

6 Reference sets

A reference set consists of a group of references to SNOMED CT components. Reference sets may optionally include additional information about each referenced component, such as language acceptability preferences.

Reference set data structures provide the fundamental pieces of the generic extensibility mechanism in RF2. They make it possible to enhance SNOMED CT-based content without modifying fundamental or core structures. They provide a mechanism that allows additional data and metadata to be attached to SNOMED CT components, as well as the ability to combine content into more usable groups. Reference sets can be used for many different purposes, for example:

- Language and dialects can be represented as Language type reference sets. The *Australian dialect reference set* is included in the SNOMED CT-AU release.
- Developing maps to and from other code systems and classifications.
- Subsets of concepts, descriptions, or relationships can be selected and presented as reference sets.

SNOMED CT-AU has a wide range of content covering various health care disciplines and clinical specialties. The *NCTIS Reference Set Library* [3] lists all the reference sets that have been developed in Australia. Additionally the methods used to create and improve existing reference sets are outlined in the *NCTIS Development Approach for Reference Sets* [4].

6.1 Reference sets – types and examples

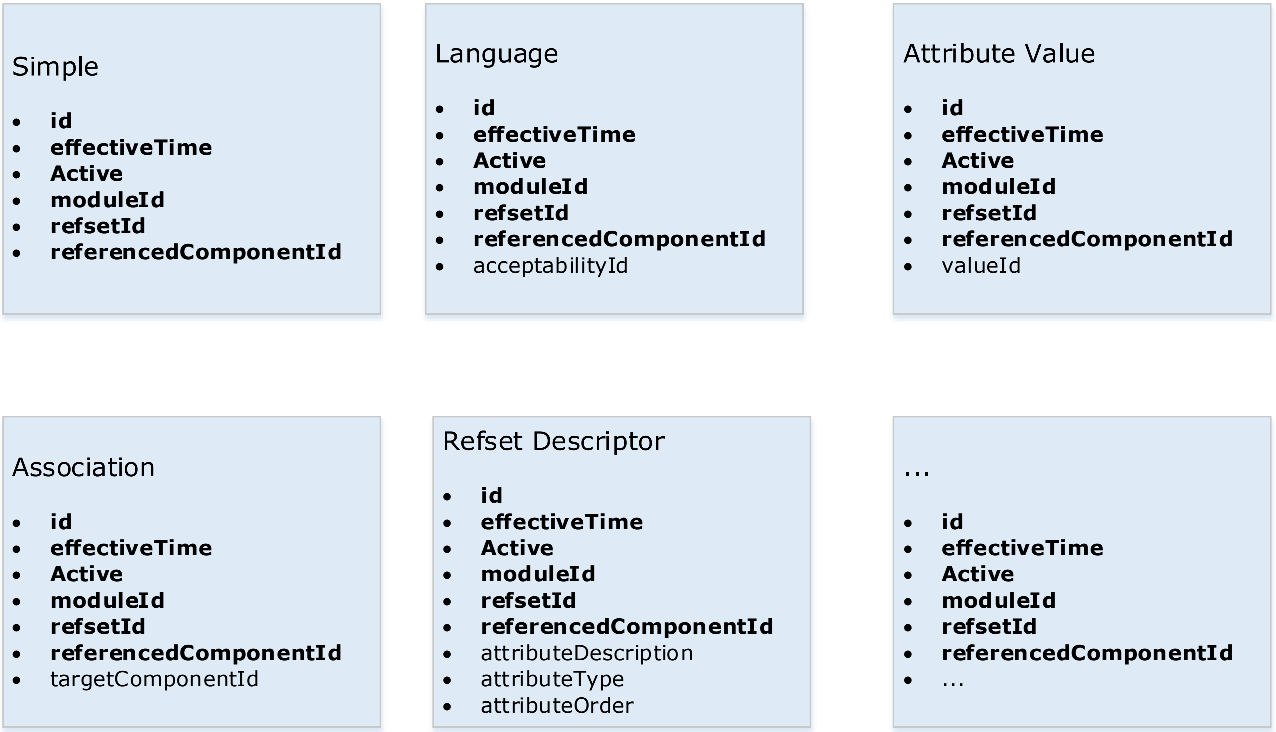
All reference sets are based on a generic data structure that is able to be extended to meet application requirements. The first four fields fulfil the same purpose as their counterparts in the core Concept, Description, and Relationship files: see Section 5.6.2 "Reference Set Specifications" of the *SNOMED TIG* [1] for related information.

Table 4: Basic reference set member format

Field	Purpose
Id	The Id provides a unique identifier for a component.
effectiveTime	The effectiveTime gives the point in time at which this version of the component came into effect.
active	The active flag states whether the component is active or inactive.
moduleId	The moduleId identifies the source module in which the component is maintained.
refSetId	Uniquely identifies the reference set that this component is a part of. It should be a descendant of *Reference set (Foundation metadata concept)*. This can be found in the *SNOMED CT Model Component (metadata)* hierarchy.
referencedComponentId	Uniquely identifies the component that this row relates to, thus defining membership of this component in the reference set.

Field	Purpose
(Optional fields)	Additional fields as required by different use cases.

As illustrated in the figure below, the first six fields listed in the table above are used in all reference sets. Additional optional fields may be appended as required to supplement the common fields and they could be of SCTID, string or integer data types. Different reference set patterns can be created by adding varying combinations of data type fields to this basic structure. Over time, NEHTA will release various types (patterns) of reference sets. This document will be extended to describe them as they are introduced; see Section 5.6 "Release Format 2 - Reference Sets Guide" of the *SNOMED TIG* for more detailed coverage of the various reference set patterns.



Simple

- **id**
- **effectiveTime**
- **Active**
- **moduleId**
- **refsetId**
- **referencedComponentId**

Language

- **id**
- **effectiveTime**
- **Active**
- **moduleId**
- **refsetId**
- **referencedComponentId**
- acceptabilityId

Attribute Value

- **id**
- **effectiveTime**
- **Active**
- **moduleId**
- **refsetId**
- **referencedComponentId**
- valueId

Association

- **id**
- **effectiveTime**
- **Active**
- **moduleId**
- **refsetId**
- **referencedComponentId**
- targetComponentId

Refset Descriptor

- **id**
- **effectiveTime**
- **Active**
- **moduleId**
- **refsetId**
- **referencedComponentId**
- attributeDescription
- attributeType
- attributeOrder

...

- **id**
- **effectiveTime**
- **Active**
- **moduleId**
- **refsetId**
- **referencedComponentId**
- ...

Figure 7: Common attributes of various reference sets

Individual reference sets are identified by a SNOMED CT-AU concept within a metadata hierarchy. This concept can be used as a link to associate metadata with the reference set using relationships to other concepts or other reference sets. The reference sets released as part of SNOMED CT-AU can be broadly categorised as follows:

- Structural reference sets
- Clinical and administrative reference sets
- Concrete domain reference sets

Additionally, reference sets may be bound to specific information models or requirements.

6.1.1 Structural reference sets

Most structural reference sets function as a mechanism for managing SNOMED CT-AU as an extension, its inherent data structures, and required release formats. These reference sets typically detail things such as historical relationships and module dependencies. Most are common to all SNOMED CT releases and extensions. Structural reference sets may be required in implementations to fulfil technical requirements. Structural reference sets are used to:

- Provide details of SNOMED CT content not detailed in the core tables (for example, the *Australian dialect reference set*).
- Describe the technical metadata associated for the release (for example, module dependencies and reference set descriptors).
- Track content changes, such as the reason for component inactivation and historical associations.

6.1.2 Clinical and administrative reference sets

A comprehensive terminology such as SNOMED CT contains concepts to cater for almost every clinical application. A challenge for implementers lies in identifying relevant terms for specific contexts. Clinical reference sets are intended to function as a means to permit more focused and specific sections of the terminology to be easily identifiable.

At their simplest, reference sets are a mechanism used to identify a subset of content from SNOMED CT-AU. An analogy would be to think of SNOMED CT-AU as a book, in which case a reference set is effectively an index entry pointing to a particular set of concepts relevant to a particular subject or use case. For example, the members of the *Pathology request test name reference set* are drawn from the *Laboratory* section within the *Procedures* hierarchy of SNOMED CT.

Clinical and administrative content reference sets are those that serve as subsets of content from SNOMED CT-AU. These are the reference sets that have the most relevance to clinicians and other users of SNOMED CT-AU. Basic subsets are produced using the Simple type reference set pattern. Most of the clinical reference sets currently released by NEHTA fall under this category. The simple type reference sets currently in SNOMED CT-AU identify a set of ConceptIds.

Table 5: Example usage of Simple type reference set

Common fields	**refSetId**	**referencedComponentId**
• *id* • *effectiveTime* • *active* • *moduleId*	*"Specimen type reference set"*	*"Urine specimen"*
• *id* • *effectiveTime* • *active* • *moduleId*	*"Specimen type reference set"*	*"Sputum specimen"*

Common fields	refSetId	referencedComponentId
• *id* • *effectiveTime* • *active* • *moduleId*	*"Specimen type reference set"*	*"Sweat specimen"*

Note: The fields in the reference set are populated by numeric identifiers, but this table displays human-readable equivalents, hence the quotation marks.

6.1.3 Concrete domain reference sets

The AMT includes a unique set of reference sets collectively referred to as concrete domain reference sets. Such reference sets are used to represent machine-readable values for numbers. These reference sets are a crucial part of the concept model and definitions, unlike most other reference sets. In the AMT they are used to represent the magnitude and units associated with medicines, as follows:

- *Strength reference set*
- *Unit of use size reference set*
- *Unit of use quantity reference set*
- *Subpack quantity reference set*

As an example, the *Strength reference set* details the strength of a product's active ingredient with both a value and units (for example, "500 milligrams") in separate fields. Representing the content as such provides a reliable mechanism for implementers to perform dose calculations and the like. The AMT is the only SNOMED CT-AU extension that currently takes advantage of concrete domains. These reference sets are described in more detail in Section 6.5.

6.1.4 Bound and non-bound reference sets

Bound reference sets are those that align with a clinical information specification and take into account data element and data group definitions, as well as other surrounding data structures, which may or may not affect the content of that reference set. The SNOMED CT concept model is also considered in this alignment process.

Non-bound reference sets are those that are agnostic of clinical information specifications, and are instead developed against a statement of purpose, scope, or general definition. Like bound reference sets, their development takes into account the SNOMED CT concept model. However, unlike bound reference sets, they do not take into account any other definitions or data items that may coexist where these reference sets might be implemented.

The reuse of bound or non-bound reference sets outside of the context within which they were developed should be approached with caution and a full analysis undertaken to ensure applicability.

Reference sets with specific bindings described by the National Clinical Terminology and Information Service (NCTIS) are categorised according to those bound to

NEHTA clinical information specification archetypes (or data groups) and those bound to other clinical information specifications.

Bound reference sets may also be developed against a very specific technical or implementation use case such as mapping to alternate codesets.

6.2 Published reference sets

The reference sets developed and released by the NCTIS fall into the categories described below.

6.2.1 Foundation and Notable concept reference sets

Foundation reference sets are those that form the basis from which all NCTIS clinical and administrative content reference sets will be developed. They will also serve as the basis for local reference set development within the SNOMED CT-AU community of practice. Sixteen Foundation reference sets have been developed, by removing content that is not applicable in Australian healthcare, such as all non-human content, as well as concepts that are not active. They are clinical or administrative reference sets that are not bound to a clinical information specification.

Similarly, seven Notable concept reference sets have been developed for the AMT that list all concepts for each of the seven notable product concepts:

- 929360061000036106 |*Medicinal product reference set*|
- 929360071000036103 |*Medicinal product unit of use reference set*|
- 929360081000036101 |*Medicinal product pack reference set*|
- 929360021000036102 |*Trade product reference set*|
- 929360031000036100 |*Trade product unit of use reference set*|
- 929360041000036105 |*Trade product pack reference set*|
- 929360051000036108 |*Containered trade product pack reference set*|

Both Foundation and Notable concept reference sets are provided to assist implementers in identifying all concepts of a given class.

6.2.2 Broad context and Intermediate reference sets

Broad context reference sets are derived from the Foundation reference sets, and are based on the terminology that is used by those working in clinical groups (for example, terminology commonly used to describe patients admitted to an orthopaedic ward). Intermediate reference sets use the same approach and contain more focused content than a Broad context reference set.

Broad context and Intermediate reference sets are suitable for use by implementers until Specific reference sets are developed, if required. In many cases, these reference sets will be the end point of development. The Broad context reference sets developed to date have been achieved by using a semi-automated method to isolate whole sections of SNOMED CT hierarchies. Again, these reference sets can be clinical or administrative and are not bound to a clinical information specification.

6.2.3 Specific reference sets

Foundation, Broad context and Intermediate reference sets can be used to create reference sets for specific implementations or instances. These Specific reference sets are bound to clinical information structures (such as data elements) or are developed to fulfil very specific definitions and use cases. Thus, Specific reference sets are only ratified for that particular use. For example, a reference set developed for use within a particular NEHTA clinical information model data element, or when a clinician would like to create a Specific reference set for use in their clinic.

6.3 Custom reference sets

Many implementers may need to create custom or refined reference sets to better suit their needs. Caution is advised if undertaking the creation of custom reference sets. In particular, the interoperability implications must be considered when exchanging standardised documents for which specific value domains are defined. In particular:

- A custom subset of an existing reference set will still allow the sending of compliant messages. However, incoming messages may potentially carry valid codes not in the custom reference set.
- A custom reference set that contains concepts additional to those specified in national exchange specifications should not include those codes in messages, as recipients of these messages may not be able to process them.
- An existing reference set may be used as the basis for a new customised reference set. For example, a copy of the *Procedure foundation reference set* may be created and customised to contain only those procedures that might be recorded in a Gastroenterology Unit. That reference set in turn could be customised for a local implementation to contain only the members required by an individual clinician (for example, "*Dr Hoffman's gastro reference set*"). This new customised reference set should be renamed accordingly and care taken not to confuse the two. This is important from both a usage and maintenance perspective, given that it is highly likely that the original reference set will be updated by NEHTA.
- Where a copy of the reference set is created and customised to satisfy local needs, validation and quality assurance would also need to be conducted locally.
- Altering the contents of a Specific reference set may invalidate it against its original scope. Sufficient analysis should be done to fully understand the impact of any proposed changes. It may be helpful to refer to the *NCTIS Development Approach for Reference Sets*, which contains definitions and development criteria for published reference sets.

There are two overall approaches to creating custom reference sets: Inclusion and Exclusion.

6.3.1 Inclusion-based customisation

A custom reference set based on inclusion principles leverages the published SNOMED CT-AU reference sets to create larger or smaller implementation reference sets. Larger reference sets may be produced by combining smaller ones, or smaller sets created by specifying extra criteria to apply to an existing reference set.

6.3.1.1 SQL examples

SNOMED CT-AU includes the 32570351000036105 |*Musculoskeletal finding reference set*|, which contains over ten thousand concepts. Yet a specific implementation for podiatry may only require concepts relating to the foot structure (56459004 |*Foot structure*|). The property 363698007 |*Finding site*| may be used to focus the restriction.

```
SELECT referencedcomponentid
FROM refset_snapshot AS MSrefset
WHERE MSrefset.refsetId = 32570351000036105
AND MSrefset.active = 1
AND MSrefset.referencedcomponentid IN
  (SELECT sourceId
   FROM relationships_snapshot
   WHERE active = 1
   AND typeId = 363698007
   AND destinationId = 56459004);
```

The result is a much smaller set of about 30 concepts. Note the query above is specifically retrieving only concepts where the finding site is 56459004 |*Foot structure*|. It would be more appropriate to use a subsumption query as described in Section 10 to include all concepts where the finding site is a type of foot structure. That is, replace "destinationId = 56459004" with a subsumption clause like "is_KindOf (destinationId, 56459004)". This query would produce around 800 concepts – all musculoskeletal findings relating to foot structures.

6.3.1.2 Exclusion-based customisation

A custom reference set may also be created by excluding certain content. This would be where the requirements of a reference set are specified and concepts that are also in another reference set are excluded. An example might be where a reference set of findings relating to pregnancy is required. A custom reference set based on just the inclusion approach might not be appropriate for use in populating a simple searchable drop-down box. The result may include a number of grouper concepts, which might be useful in a hierarchical navigation menu, but are not concepts an end user would want to select from when presented in a flat list. However, these can be excluded by using the *Clinical finding grouper exclusion reference set.*

Note: The *Clinical finding grouper exclusion reference set* is a subset of SNOMED CT-AU content that has been identified as inappropriate for recording in a patient's medical records; typically these are groupers of insufficient specificity to be of use in patient care. Depending on the navigation and recording interfaces used, this reference set may or may not be useful.

6.3.1.3 SQL examples

Using the example above where a reference set of pregnancy findings is needed, the requirements might be specified as all concepts that are types of 118185001 |*Finding related to pregnancy*| are included. Assuming a subsumption function (as described in Section 10) such as "is_KindOf (candidate,supertype)", such a query may be specified as:

```
SELECT id
FROM concepts_snapshot
WHERE is_KindOf(id,118185001);
```

This query produces around 1,704 concepts[3] – including some that a clinician would not want to include in a patient record. To exclude these, the query can be amended to restrict concepts that are also in the *Clinical finding grouper exclusion reference set*:

```
SELECT id
FROM concepts_snapshot
WHERE is_KindOf(id, 118185001)
AND id NOT IN
  (SELECT referencedconceptid
   FROM refset_snapshot
   WHERE refsetId = 171991000036103);
```

The results from this query now only include 1,621 concepts. That is, 83 grouper concepts have been excluded. Some of the excluded concepts are as follows:

- 289723002 |*Finding of duration of uterine contraction*|
- 366329008 |*Speed of delivery of placenta – finding*|
- 408827003 |*Antenatal HIV blood screening test status*|
- 118212000 |*Parity finding*|
- 118185001 |*Finding related to pregnancy*|

6.4 Australian dialect reference set (ADRS)

The ADRS is a language type reference set; it specifies the Preferred Terms and acceptable Synonyms identified as appropriate for the recording of clinical information in Australian e-health implementations. The first release of the ADRS was derived from the UK dialect subset that is released by the IHTSDO as part of the SNOMED CT International Release. Subsequent development of the ADRS continues to refine the reference set by addressing content in SNOMED CT that is inappropriate for Australian usage and will include additional Australian expressions and spellings.

Implementation of the ADRS is required, so that the Preferred Term for each concept may be identified. SNOMED CT-AU provides two types of descriptions:

- Fully Specified Names – the true meaning of the concept.
- Synonyms – other names for a concept that are useful in a variety of settings.

Language type reference sets in SNOMED CT Release Format 2 (RF2) annotate RF2 Synonym descriptions with one of three mutually exclusive values to provide localisation:

Preferred Indicates that Synonyms annotated with this value are the preferred way of describing the concept (also known as the

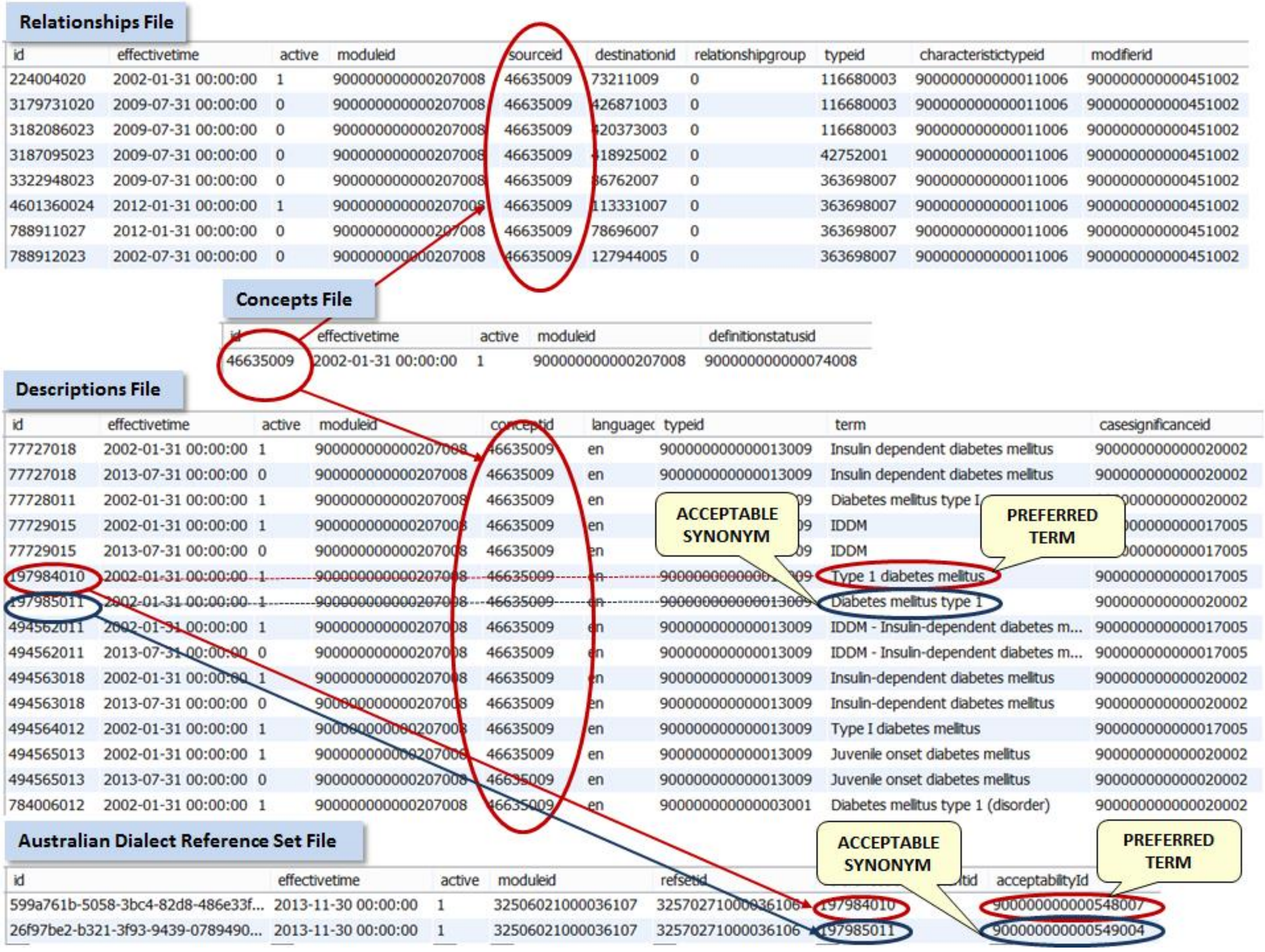
[3] Based on SNOMED CT-AU November 2013 data. Different releases may produce different results.

"Preferred Term"). A Language type reference set is required to have exactly one active preferred Synonym for each concept.

Acceptable Indicates that Synonyms annotated with this value are not the preferred way of describing a concept, but are acceptable in the context of the Language type reference set.

Not Acceptable Indicates that the Synonym is not an acceptable way to describe the concept in this Language type reference set's context. Language type reference sets do not explicitly reference Synonyms with this value; "Not Acceptable" is instead implied by the absence of a reference to those synonyms.

A query to extract the Preferred Term for a particular concept would span the various component files as illustrated in the figure below.



Relationships File

id	effectivetime	active	moduleid	sourceid	destinationid	relationshipgroup	typeid	characteristictypeid	modifierid
224004020	2002-01-31 00:00:00	1	900000000000207008	46635009	73211009	0	116680003	900000000000011006	900000000000451002
3179731020	2009-07-31 00:00:00	0	900000000000207008	46635009	426871003	0	116680003	900000000000011006	900000000000451002
3182086023	2009-07-31 00:00:00	0	900000000000207008	46635009	420373003	0	116680003	900000000000011006	900000000000451002
3187095023	2009-07-31 00:00:00	0	900000000000207008	46635009	418925002	0	42752001	900000000000011006	900000000000451002
3322948023	2009-07-31 00:00:00	0	900000000000207008	46635009	36762007	0	363698007	900000000000011006	900000000000451002
4601360024	2012-01-31 00:00:00	1	900000000000207008	46635009	113331007	0	363698007	900000000000011006	900000000000451002
788911027	2012-01-31 00:00:00	0	900000000000207008	46635009	78696007	0	363698007	900000000000011006	900000000000451002
788912023	2002-07-31 00:00:00	0	900000000000207008	46635009	127944005	0	363698007	900000000000011006	900000000000451002

Concepts File

id	effectivetime	active	moduleid	definitionstatusid
46635009	2002-01-31 00:00:00	1	900000000000207008	900000000000074008

Descriptions File

id	effectivetime	active	moduleid	conceptid	languagec	typeid	term	casesignificanceid
77727018	2002-01-31 00:00:00	1	900000000000207008	46635009	en	900000000000013009	Insulin dependent diabetes mellitus	900000000000020002
77727018	2013-07-31 00:00:00	0	900000000000207008	46635009	en	900000000000013009	Insulin dependent diabetes mellitus	900000000000020002
77728011	2002-01-31 00:00:00	1	900000000000207008	46635009	en	09	Diabetes mellitus type I	00000000020002
77729015	2002-01-31 00:00:00	1	900000000000207008	46635009	en	09	IDDM	00000000017005
77729015	2013-07-31 00:00:00	0	900000000000207008	46635009	en	09	IDDM	00000000017005
197984010	2002-01-31 00:00:00	1	900000000000207008	46635009	en	9000000000000 09	Type 1 diabetes mellitus	900000000000017005
197985011	2002-01-31 00:00:00	1	900000000000207008	46635009	en	900000000000013009	Diabetes mellitus type 1	900000000000020002
494562011	2002-01-31 00:00:00	1	900000000000207008	46635009	en	900000000000013009	IDDM - Insulin-dependent diabetes m...	900000000000017005
494562011	2013-07-31 00:00:00	0	900000000000207008	46635009	en	900000000000013009	IDDM - Insulin-dependent diabetes m...	900000000000017005
494563018	2002-01-31 00:00:00	1	900000000000207008	46635009	en	900000000000013009	Insulin-dependent diabetes mellitus	900000000000020002
494563018	2013-07-31 00:00:00	0	900000000000207008	46635009	en	900000000000013009	Insulin-dependent diabetes mellitus	900000000000020002
494564012	2002-01-31 00:00:00	1	900000000000207008	46635009	en	900000000000013009	Type I diabetes mellitus	900000000000017005
494565013	2002-01-31 00:00:00	1	900000000000207008	46635009	en	900000000000013009	Juvenile onset diabetes mellitus	900000000000020002
494565013	2013-07-31 00:00:00	0	900000000000207008	46635009	en	900000000000013009	Juvenile onset diabetes mellitus	900000000000020002
784006012	2002-01-31 00:00:00	1	900000000000207008	46635009	en	900000000000003001	Diabetes mellitus type 1 (disorder)	900000000000020002

Australian Dialect Reference Set File

id	effectivetime	active	moduleid	refsetid	tid	acceptabilityId
599a761b-5058-3bc4-82d8-486e33f...	2013-11-30 00:00:00	1	32506021000036107	32570271000036106	197984010	900000000000548007
26f97be2-b321-3f93-9439-0789490...	2013-11-30 00:00:00	1	32506021000036107	32570271000036106	197985011	900000000000549004

Figure 8: Retrieving the Preferred Term and acceptable Synonym for concept 46635009

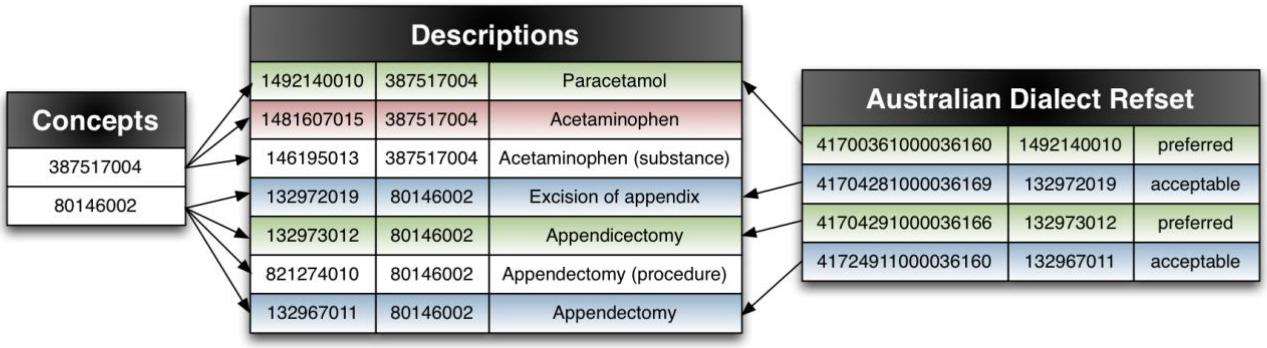
6.4.1 Technical summary

The ADRS is a component reference set as described in Section 5.6.2.8 "Language Reference Set" of the *SNOMED TIG*. As with all reference sets, the **referencedComponentId** is the component being referenced. For Language type reference sets, this field contains an Id from the Descriptions file. The last field, **acceptabilityId**, indicates the preference with the current range of values being:

- 900000000000548007 |*Preferred*|

- 900000000000549004 |*Acceptable*|

Using both the Descriptions file and the ADRS, it is possible to determine which descriptions are considered to be preferred. Each active concept has exactly one preferred description.



Concepts
387517004
80146002

Descriptions		
1492140010	387517004	Paracetamol
1481607015	387517004	Acetaminophen
146195013	387517004	Acetaminophen (substance)
132972019	80146002	Excision of appendix
132973012	80146002	Appendicectomy
821274010	80146002	Appendectomy (procedure)
132967011	80146002	Appendectomy

Australian Dialect Refset		
41700361000036160	1492140010	preferred
41704281000036169	132972019	acceptable
41704291000036166	132973012	preferred
41724911000036160	132967011	acceptable

Figure 9: Relationship between the concepts, descriptions and Language type reference set

The figure above shows an example taken from SNOMED CT-AU, showing how the ADRS expresses that:

- "Paracetamol" is preferred (green) and "Acetaminophen" is not acceptable (red).
- "Appendicectomy" is preferred (green), while "Appendectomy" is acceptable (blue), as is "Excision of appendix".

That is, the Language type reference set can be joined with the content of the Descriptions file to determine the preferred and acceptable Synonyms for a concept or concepts.

Note: Fully Specified Names are not referenced – they are always acceptable and preference is irrelevant. Note also that columns have been omitted from the example for brevity.

6.4.1.1 SQL examples

Consider the conceptId 46635009. All available descriptions can be retrieved using the following query:

```
SELECT conceptId, id ,term
FROM descriptions_snapshot
WHERE conceptId = 46635009;
```

The acceptability of each description can be identified through the ADRS by extending the query as:

```
SELECT conceptId,D.id,term, acceptabilityid
FROM descriptions_snapshot AS D
LEFT JOIN language_refset_snapshot AS ADRS
ON D.id = ADRS.referencedcomponentid
WHERE D.conceptId = 46635009;
```

Note: In the above query, using a **LEFT JOIN** results in null "acceptabilityId" values for descriptions not referenced in the ADRS. An **INNER JOIN** will return only descriptions that are actually referenced.

Ultimately, the current Preferred Term for the concept 387517004 can be retrieved with:

```
SELECT conceptId,D.id,term, acceptabilityid
FROM descriptions_snapshot AS D
INNER JOIN language_refset_snapshot AS ADRS
ON D.id = ADRS.referencedcomponentid
WHERE D.conceptId = 46635009
AND ADRS.acceptabilityid = 900000000000548007 -- ConceptId for 'Preferred'
AND ADRS.active = 1
```

Note: It is necessary to apply the "ADRS.active = 1" filter to identify the current Preferred Term, as this can change over time.

To improve readability, developers may prefer to create a custom function or method that accepts a conceptId as a parameter and returns the relevant Preferred Term. For example:

```
delimiter //
DROP FUNCTION IF EXISTS get_PT
//
CREATE FUNCTION get_PT(candidate bigint(20)) RETURNS varchar(300)

BEGIN
RETURN (SELECT term
FROM descriptions_snapshot AS D
INNER JOIN language_refset_snapshot AS ADRS
ON D.id = ADRS.referencedcomponentid
WHERE D.conceptId = candidate
AND ADRS. acceptabilityid = 900000000000548007
AND ADRS.active = 1);
END
//
```

This function can then be executed whenever a Preferred Term is required as follows:

```
SELECT get_PT(64459004);
```

6.4.2 Implementation suggestions

Retrieving the Preferred Terms for concepts is likely to be the most frequently undertaken terminology task within any implementation. Some performance issues may be experienced if the nested query illustrated above is repeatedly executed. To mitigate this, implementers may choose to store denormalised data structures. For the ADRS, this involves creating a precomputed table of conceptIds and their associated preferred descriptions.

Alternatively, index optimisation alone may provide sufficient performance gains.

6.5 Working with concrete domains reference sets

Concrete domains and data type properties are a way of including concrete data values as defining attributes of concepts. The phrase "concrete domains" stems from research in the late 1970s into mathematical descriptions of the semantics of programming languages in an effort to distinguish data from semantic domains. This same phrase is commonly used when describing these capabilities of description logic.

While the relationships in SNOMED CT allow a concept to have an attribute with a concept value (e.g. "21433011000036107 |*paracetamol*|*")*, data type properties provide attributes that have a concrete data value (e.g. "5").

The AMT only uses the concrete domains of real numbers and integers. These are used to represent the defining numeric attributes of AMT concepts, specifically:

- *Strength*
  - Such as the concentration of an ingredient.
- *Unit of use size*
  - Such as the volume of a medicine in an ampoule.
- *Unit of use quantity*
  - Such as the number of tablets in a pack.
- *Subpack quantity*
  - For example, the number of identical components within a primary pack containing subpacks.

The table below describes fields within reference set files that are used for concrete domains.

Table 6: RF2 concrete domain reference set file format

Field	Data type	Purpose
referencedComponentId	SctId	The component with which the concrete value is to be associated. In the context of the AMT this is always a relationship identifier.
unitId	SctId	A child of 258666001 |*Unit*|. In the AMT this is always a type of the AMT[4] |*Unit of measure*| concept.
operatorId	SctId	A child of |*Operator id value*| in the metadata hierarchy. Currently in the AMT this field will always hold the value 700000051000036108 |*Equal to*|. Other values may be introduced into the AMT in the future to support ranges if necessary (such as "greater than" and "less than").
value	String	The concrete value to be associated with the referenced component. The precise data type presented in this string is dependent upon the particular subtype of the |*Concrete domain type*| concept, which will also be reflected in the Specific reference set's descriptor. For the AMT this value is a double precision floating number for the *Strength reference set*, *Unit of use size reference set*, and *Unit of*

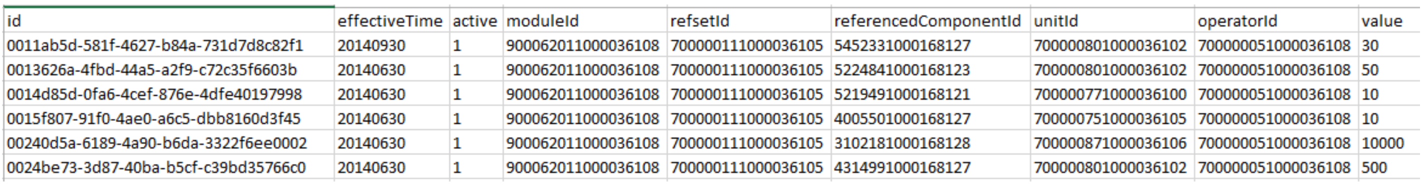
[4] As distinct from being a child of the SNOMED CT |*Unit of measure*| concept.

Field	Data type	Purpose
		use quantity reference set. For the *Subpack quantity reference set* this value will be an integer.

6.5.1 Strength reference set

The *Strength reference set* includes numerical datatype properties used in the definition of MPUU concepts.

The following figure depicts several members of the *Strength reference set* (from the January 2016 Snapshot release).



id	effectiveTime	active	moduleId	refsetId	referencedComponentId	unitId	operatorId	value
0011ab5d-581f-4627-b84a-731d7d8c82f1	20140930	1	900062011000036108	700000111000036105	5452331000168127	700000801000036102	700000051000036108	30
0013626a-4fbd-44a5-a2f9-c72c35f6603b	20140630	1	900062011000036108	700000111000036105	5224841000168123	700000801000036102	700000051000036108	50
0014d85d-0fa6-4cef-876e-4dfe40197998	20140630	1	900062011000036108	700000111000036105	5219491000168121	700000771000036100	700000051000036108	10
0015f807-91f0-4ae0-a6c5-dbb8160d3f45	20140630	1	900062011000036108	700000111000036105	4005501000168127	700000751000036105	700000051000036108	10
00240d5a-6189-4a90-b6da-3322f6ee0002	20140630	1	900062011000036108	700000111000036105	3102181000168128	700000871000036106	700000051000036108	10000
0024be73-3d87-40ba-b5cf-c39bd35766c0	20140630	1	900062011000036108	700000111000036105	4314991000168127	700000801000036102	700000051000036108	500

Figure 10: Sample of the Strength reference set

Specifically, members of the *Strength reference set* refer to HAS AUSTRALIAN BoSS relationships and add a property[5] including:

- a value (for example, 500);
- a reference to the unit of measure for the value (for example, 700000801000036102 |*mg/each*|); and
- a reference to an operator, at present only 700000051000036108 |*Equal to*|.

This further qualifies the HAS AUSTRALIAN BoSS relationship by specifying the quantity of the ingredient included – from the example above "equal to 500 mg/each".

Note that the strength of the intended active ingredient is expressed in terms of the Australian BoSS, not the intended active ingredient itself (which may differ). For example a product may be defined as follows:

- HAS AUSTRALIAN BoSS |*morphine sulfate*|;
- HAS INTENDED ACTIVE INGREDIENT |*morphine*|; and
- Strength of "10 mg/each".

Concentration based strengths and those with numerator and denominator components are represented using composite units of measure, such as "mg/mL" or "mg/each". Medicinal concept Preferred Terms may still use the most preferred human representation, however the *Strength reference set* will always represent normalised strength values using a denominator of one. For example "50 mg/5 mL" might be the human-readable strength, whereas the *Strength reference set* will contain "10 mg/mL".

[5] That is, the reference set member is an annotation of the "**referencedComponentId**" field with additional properties.

While the *Strength reference set* will typically represent normalised strength values, there are some exceptions (mainly for patches) whereby the strength value and units represent the relevant release rate (as this is the clinically relevant attribute).

6.5.1.1 Normalised unit of measure for products with same BoSS but different unit

Where different MPUU concepts share the same BoSS (that is, have the same destination *Substance* for the HAS AUSTRALIAN BoSS relationship), and different *Composite units of measure* are used (say “microgram/mL” and “mg/mL”), the *Composite unit of measure* is normalised to only a single unit (for example, only to “microgram/mL”). This is to allow easier strength comparisons across similar products.

While the normalised strength in the *Strength reference set* may be “microgram/mL”, the denormalised human friendly strength as displayed within the concept’s Preferred Terms may still be “mg/mL” as per clinical documentation.

6.5.1.2 Products with no BoSS strength

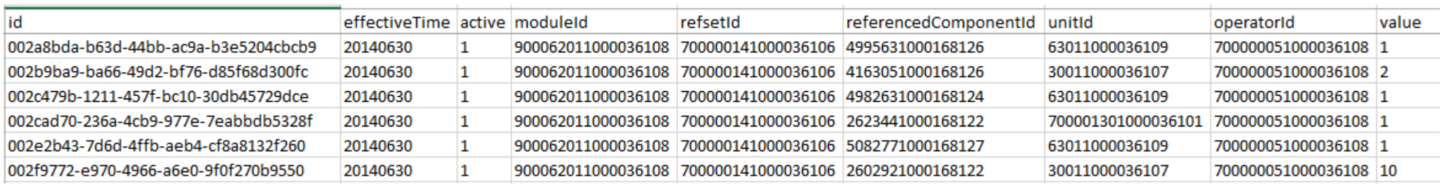
There is a strength associated with every HAS AUSTRALIAN BoSS relationship. Products where no strength logically exists for the ingredient will not have a HAS AUSTRALIAN BoSS relationship and hence no associated *Strength reference set* member. Some examples of AMT products with no *Strength* attribute are:

- Foods/nutritional supplements (such as vitamins, minerals and trace elements with carbohydrate).
- Diagnostic strips (for example, glucose indicator blood).
- Nonmedicated dressings/bandages (such as bandage tubular; bandage retention cohesive heavy).
- Inert substances, diluents.

6.5.2 Unit of use size reference set

The *Unit of use size reference set* specifies the quantity of an MPUU that constitutes a unit of use. This is usually the administrable dose unit such as “1 tablet” or “5 mL”. However, for continuous products like creams, a measurable and administrable dose unit does not exist.

The following figure depicts several members of the *Unit of use size reference set* (from the January 2016 Snapshot release).



id	effectiveTime	active	moduleId	refsetId	referencedComponentId	unitId	operatorId	value
002a8bda-b63d-44bb-ac9a-b3e5204cbcb9	20140630	1	900062011000036108	700000141000036106	4995631000168126	63011000036109	700000051000036108	1
002b9ba9-ba66-49d2-bf76-d85f68d300fc	20140630	1	900062011000036108	700000141000036106	4163051000168126	30011000036107	700000051000036108	2
002c479b-1211-457f-bc10-30db45729dce	20140630	1	900062011000036108	700000141000036106	4982631000168124	63011000036109	700000051000036108	1
002cad70-236a-4cb9-977e-7eabbdb5328f	20140630	1	900062011000036108	700000141000036106	2623441000168122	700001301000036101	700000051000036108	1
002e2b43-7d6d-4ffb-aeb4-cf8a8132f260	20140630	1	900062011000036108	700000141000036106	5082771000168127	63011000036109	700000051000036108	1
002f9772-e970-4966-a6e0-9f0f270b9550	20140630	1	900062011000036108	700000141000036106	2602921000168122	30011000036107	700000051000036108	10

Figure 11: Sample of the Unit of use size reference set

For MPUU concepts with premeasured, measured or indivisible forms, this reference set will simply indicate a single unit of the dose form. For example the unit of use size for a |*paracetamol 500 mg tablet*| is “1 tablet”. Other examples are “1 patch” or “1 suppository”.

For concentration-based products, the unit of use size is the total quantity of the medicine in the MPUU concept. For example |*benztropine mesylate 2 mg/2 mL injection, ampoule*| has its strength in the *Strength reference set* as "1 mg/mL" and in the *Unit of use size reference set* its value is "2 mL".

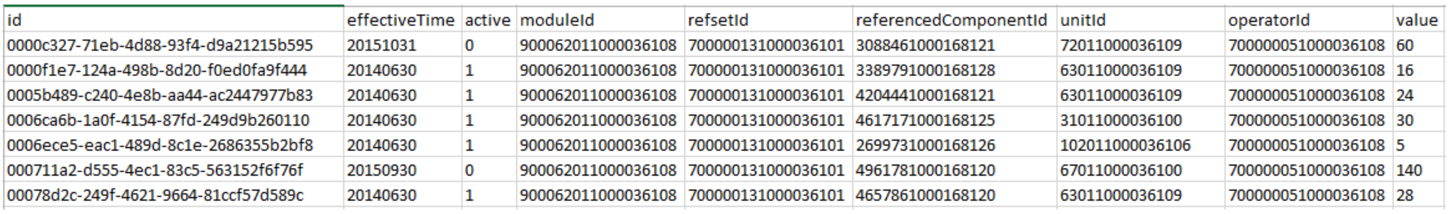
The combination of the *Unit of use size reference set* and the *Strength reference set* provide the ability to calculate the total ingredient quantity for a unit.

All MPUU concepts will have an entry in the *Unit of use size reference set*. It is not possible to sensibly provide a unit of use size as a precise amount for continuous form products such as solutions, creams, ointments or some inhalations. These products will have a unit of use size of "1 each" intended to ease total pack quantity calculations by consistently providing a value.

6.5.3 Unit of use quantity reference set

The *Unit of use quantity reference set* includes numerical datatype properties used in the definition of Medicinal Product Pack (MPP) and TPP concepts. The unit of use quantity represents the pack quantity for a given medicinal or trade pack concept. Some examples are "100 capsule", "20 tube", "30 sachet" and "10 vial".

The following figure depicts several members of the *Unit of use quantity reference set* (from the January 2016 Snapshot files).



id	effectiveTime	active	moduleId	refsetId	referencedComponentId	unitId	operatorId	value
0000c327-71eb-4d88-93f4-d9a21215b595	20151031	0	900062011000036108	700000131000036101	3088461000168121	72011000036109	700000051000036108	60
0000f1e7-124a-498b-8d20-f0ed0fa9f444	20140630	1	900062011000036108	700000131000036101	3389791000168128	63011000036109	700000051000036108	16
0005b489-c240-4e8b-aa44-ac2447977b83	20140630	1	900062011000036108	700000131000036101	4204441000168121	63011000036109	700000051000036108	24
0006ca6b-1a0f-4154-87fd-249d9b260110	20140630	1	900062011000036108	700000131000036101	4617171000168125	31011000036100	700000051000036108	30
0006ece5-eac1-489d-8c1e-2686355b2bf8	20140630	1	900062011000036108	700000131000036101	2699731000168126	102011000036106	700000051000036108	5
000711a2-d555-4ec1-83c5-563152f6f76f	20150930	0	900062011000036108	700000131000036101	4961781000168120	67011000036100	700000051000036108	140
00078d2c-249f-4621-9664-81ccf57d589c	20140630	1	900062011000036108	700000131000036101	4657861000168120	63011000036109	700000051000036108	28

Figure 12: Sample of the Unit of use quantity reference set

Members of the *Unit of use quantity reference set* refer to HAS MPUU or HAS TPUU relationships between MPP/TPP concepts and MPUU/TPUU concepts and add a property including:

- an integer value (for example, 30);
- a reference to the unit of measure (for example, 154011000036109 |*tablet*|); and
- a reference to an operator, at present only 700000051000036108 |*Equal to*|.

This has the effect of quantifying the HAS MPUU or HAS TPUU relationship to specify how many of each type of MPUU/TPUU concepts an MPP/TPP concept contains. In the example above this would be "equal to 30 tablets".

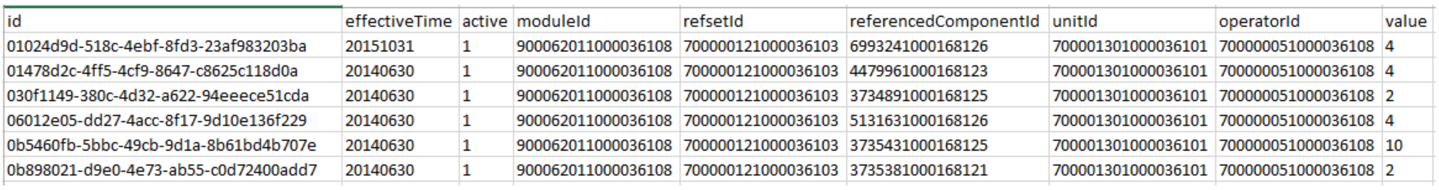
Every active HAS MPUU relationship between an MPP concept and an MPUU concept will be referred to by a single active member of the *Unit of use quantity reference set*. Similarly, every active HAS TPUU relationship between a TPP concept and a TPUU concept will be referred to by a single active member of the *Unit of use quantity reference set*.

6.5.4 Subpack quantity reference set

The *Subpack quantity reference set* quantifies the HAS SUBPACK relationship. An MPP containing subpacks has multiple identical subpack MPPs. That is, for each HAS SUBPACK relationship, the *Subpack quantity reference set* specifies how many of the subpack MPPs are included in the *Medicinal product pack* concept containing the subpacks.

Similarly, the reference set quantifies the HAS SUBPACK relationship from a CTPP pack containing subpacks with each subpack CTPP. This specifies how many of the subpack CTPPs are included in the CTPP pack containing subpacks.

The following figure depicts several members of the *Subpack quantity reference set* (from the January 2016 Snapshot release).



id	effectiveTime	active	moduleId	refsetId	referencedComponentId	unitId	operatorId	value
01024d9d-518c-4ebf-8fd3-23af983203ba	20151031	1	900062011000036108	700000121000036103	6993241000168126	700001301000036101	700000051000036108	4
01478d2c-4ff5-4cf9-8647-c8625c118d0a	20140630	1	900062011000036108	700000121000036103	4479961000168123	700001301000036101	700000051000036108	4
030f1149-380c-4d32-a622-94eeece51cda	20140630	1	900062011000036108	700000121000036103	3734891000168125	700001301000036101	700000051000036108	2
06012e05-dd27-4acc-8f17-9d10e136f229	20140630	1	900062011000036108	700000121000036103	5131631000168126	700001301000036101	700000051000036108	4
0b5460fb-5bbc-49cb-9d1a-8b61bd4b707e	20140630	1	900062011000036108	700000121000036103	3735431000168125	700001301000036101	700000051000036108	10
0b898021-d9e0-4e73-ab55-c0d72400add7	20140630	1	900062011000036108	700000121000036103	3735381000168121	700001301000036101	700000051000036108	2

Figure 13: Sample of the Subpack quantity reference set

For example the HAS SUBPACK relationship between the subpack MPP:

- 28347011000036104 |*levonorgestrel 50 microgram + ethinyloestradiol 30 microgram tablet [6] (&) levonorgestrel 75 microgram + ethinyloestradiol 40 microgram tablet [5] (&) levonorgestrel 125 microgram + ethinyloestradiol 30 microgram tablet [10] (&) inert substance tablet [7], 28*|

and the MPP pack containing subpacks:

- 27070011000036109 |*levonorgestrel 50 microgram + ethinyloestradiol 30 microgram tablet [6] (&) levonorgestrel 75 microgram + ethinyloestradiol 40 microgram tablet [5] (&) levonorgestrel 125 microgram + ethinyloestradiol 30 microgram tablet [10] (&) inert substance tablet [7], 4 x 28*|

is referred to by the *Subpack quantity reference set* with the value "4 each", indicating that the MPP pack containing subpacks contains four of the subpack MPP instances. Each CTPP descendant of an MPP pack containing subpacks will have a HAS SUBPACK relationship to a CTPP which is a descendant of the parent MPP's subpack, and the specified subpack quantity (in this case four) will match the subpack quantity specified for the MPP parent.

6.6 Mapping reference sets

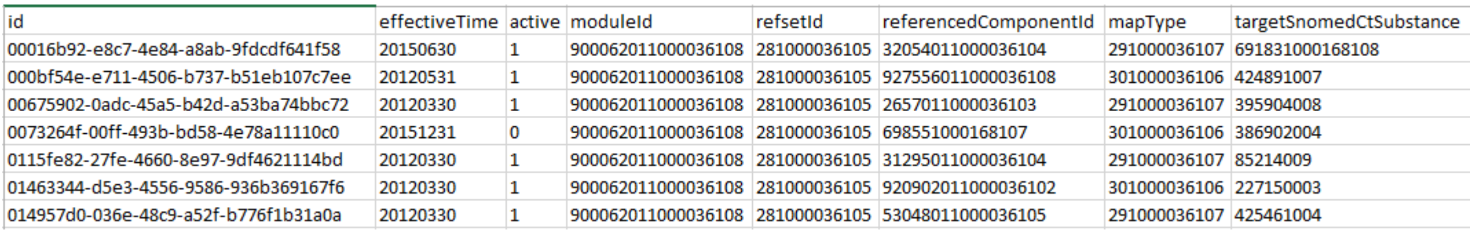
There are currently two mapping reference sets in the AMT. These are customised reference sets and therefore do not conform to any current RF2 reference set patterns. The files provide a mapping between AMT concepts and SNOMED CT-AU terminology, and between the AMT and other code systems.

6.6.1 Substance to SNOMED CT-AU mapping reference set

The *Substance to SNOMED CT-AU mapping reference set* is developed for the implementers of the AMT, SNOMED CT-AU and NEHTA Detailed Clinical Models[6] to enable use of AMT content integrated with SNOMED CT.

The AMT's *Medicinal substance* concepts represent a parallel set of substance concepts overlapping meaning with parts of the *Substance* hierarchy in SNOMED CT. As this hierarchy is undergoing review and likely to change in the future, AMT products reference the AMT *Medicinal substances* as ingredients rather than SNOMED CT concepts. The *Substance to SNOMED CT-AU mapping reference set* will contain all the AMT substances that are used in a modelled AMT product with a corresponding equivalent or supertype[7] map to a substance in SNOMED CT-AU.

The following figure depicts several members of the AMT's *Substance to SNOMED CT-AU mapping reference set* (from the January 2016 Snapshot release).



id	effectiveTime	active	moduleId	refsetId	referencedComponentId	mapType	targetSnomedCtSubstance
00016b92-e8c7-4e84-a8ab-9fdcdf641f58	20150630	1	900062011000036108	281000036105	32054011000036104	291000036107	691831000168108
000bf54e-e711-4506-b737-b51eb107c7ee	20120531	1	900062011000036108	281000036105	927556011000036108	301000036106	424891007
00675902-0adc-45a5-b42d-a53ba74bbc72	20120330	1	900062011000036108	281000036105	2657011000036103	291000036107	395904008
0073264f-00ff-493b-bd58-4e78a11110c0	20151231	0	900062011000036108	281000036105	698551000168107	301000036106	386902004
0115fe82-27fe-4660-8e97-9df4621114bd	20120330	1	900062011000036108	281000036105	31295011000036104	291000036107	85214009
01463344-d5e3-4556-9586-936b369167f6	20120330	1	900062011000036108	281000036105	920902011000036102	301000036106	227150003
014957d0-036e-48c9-a52f-b776f1b31a0a	20120330	1	900062011000036108	281000036105	53048011000036105	291000036107	425461004

Figure 14: Sample of the Substance to SNOMED CT-AU mapping reference set

The **referencedComponentId** describes the source AMT *Substance* concept and the **targetSnomedCtSubstance** describes the target SNOMED CT-AU *Substance* concept that is mapped to it.

The **mapType** field describes the mapping correlation between the two identifiers:

- 291000036107 |*Equivalent*| – the AMT and SNOMED CT-AU *Substance* concepts are equivalently mapped; or
- 301000036106 |*Generalise*| – the AMT *Substance* is mapped to a supertype SNOMED CT-AU concept (that is, the AMT *Substance* is more specific than the SNOMED CT-AU *Substance*).

Further details of the mapping correlation are as follows:

- Equivalent maps are bi-directional and indicate that the AMT *Substance* concept identified is semantically equivalent to the specified SNOMED CT *Substance* concept.
  - For example, the AMT substance 2425011000036109 |*oxazepam*| is equivalent to the SNOMED CT-AU substance 387455006 |*Oxazepam*|.
- Supertype maps of AMT substances that have no equivalent SNOMED CT-AU substances are mapped to the nearest parent concept (that is, supertype concept) in the SNOMED CT-AU *Substance* hierarchy. This is a uni-directional map and must only be used from the AMT to SNOMED CT-AU.
  - For example, the AMT substance 927556011000036108 |*meningococcal group W135 conjugate vaccine*| is more specific than

[6] Further information on NEHTA Detailed Clinical Models can be obtained from the following link http://www.nehta.gov.au/implementation-resources/clinical-documents/detailed-clinical-model-library.

[7] That is, the nearest relevant parent concept.

any SNOMED CT-AU substance concept found (at the time of mapping), and is therefore mapped to the nearest SNOMED CT-AU supertype substance concept of 424891007 |*Meningococcus vaccine*|.

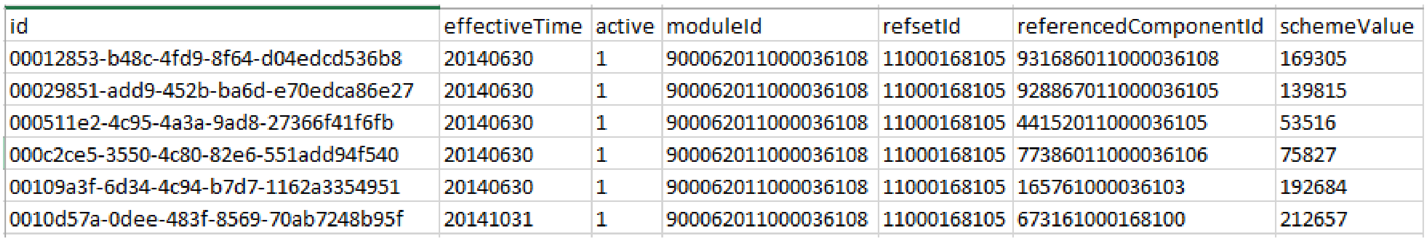
Note: Use of the map to integrate the AMT with SNOMED CT substances may result in incorrect concept inferences. This is due to known subtype relationship issues (IS A overloading) in the SNOMED CT *Substance* hierarchy, which is being addressed by an IHTSDO project. However, until this work is completed and the SNOMED CT International edition is updated, care must be taken using the AMT *Substance to SNOMED CT-AU mapping reference set*.

6.6.2 Australian Register of Therapeutic Goods Identifier (ARTG Id) reference set

The ARTG Id is the primary identifier for therapeutic goods included in the TGA's Australian Register of Therapeutic Goods (ARTG). It is intended to support implementers for mapping purposes and identification of products.

The *ARTG Id reference set* allows ARTG Ids (as a string) to be optionally associated with one or more CTPP concepts. The ARTG Id is an identifier from an external identifier scheme (that is, not native to SNOMED CT) and is therefore a non-defining attribute of the CTPP class.

The following figure depicts several members of the AMT's *ARTG ID reference set* (from the January 2016 Snapshot release).



id	effectiveTime	active	moduleId	refsetId	referencedComponentId	schemeValue
00012853-b48c-4fd9-8f64-d04edcd536b8	20140630	1	900062011000036108	11000168105	931686011000036108	169305
00029851-add9-452b-ba6d-e70edca86e27	20140630	1	900062011000036108	11000168105	928867011000036105	139815
000511e2-4c95-4a3a-9ad8-27366f41f6fb	20140630	1	900062011000036108	11000168105	44152011000036105	53516
000c2ce5-3550-4c80-82e6-551add94f540	20140630	1	900062011000036108	11000168105	77386011000036106	75827
00109a3f-6d34-4c94-b7d7-1162a3354951	20140630	1	900062011000036108	11000168105	165761000036103	192684
0010d57a-0dee-483f-8569-70ab7248b95f	20141031	1	900062011000036108	11000168105	673161000168100	212657

Figure 15: Sample of the ARTG ID reference set

The **referencedComponentId** describes a CTPP concept that has some associated ARTG Id while the schemeValue describes the ARTG Id identifier string that is mapped to the CTPP.

The following ARTG Id maps are possible:

- One ARTG Id is associated with a single CTPP. This accounts for the vast majority of products.
  - For example, ARTG Id 75592 maps to the CTPP concept 20299011000036105 |*Diclocil 500 mg capsule, 24, blister pack*|.
- One ARTG Id is associated with multiple CTPPs. This occurs typically for products that are marketed in multiple, differing pack quantities.
  - An example is ARTG Id 71816, which maps to two CTPP concepts:
    - 20104011000036108 |*Aciclovir (GenRx) 200 mg tablet, 50, blister pack*|; and
    - 20105011000036107 |*Aciclovir (GenRx) 200 mg tablet, 90, blister pack*|.

    - The above ARTG Id is present in two separate lines in the reference set each citing different CTPP concepts.
- Multiple ARTG Ids are associated with a single CTPP. This can occur when the same product obtains a new ARTG Id and the previous ARTG Id remains in the source TGA data.
- Even if the old ARTG Id is deprecated and removed from the ARTG, the AMT continues to represent the previous ARTG Id because this information may be used to record historical data.
    - For example, the ARTG Ids 120662 and 77830 are mapped to the same CTPP concept 18285011000036108 |*Seretide MDI 125/25 inhalation: pressurised, 120 actuations, metered dose aerosol can*|.
        - The higher ARTG Id value of 120662 reflects the most current ARTG Id and should be used for any mapping or matching purposes.
        - The lower ARTG Id value of 77830 may be used to record or match on some past medication history.
    - The above two ARTG Ids are present in two separate lines in the reference set, each citing the same CTPP.

Products with no ARTG Id as assigned by the TGA do not have an entry in this file. Such products include RPBS products that do not undergo TGA registration, for example non-medicated dressings, diagnostic strips and nutritional supplements.

7 Implementation approaches

This section discusses three generic scenarios for Australian terminology implementations. Although many scenarios are possible, the principles described here should be generally applicable.

7.1 Mapping implementation

Scenario: Existing system with a local terminology in need of external interoperability.

A common scenario will be one where systems that currently use local or proprietary terminologies and require interoperability with external SNOMED CT-AU or AMT-based systems.

In this case, the implementer may prefer to use SNOMED CT-AU or the AMT only on the boundary of their system, as a basis for generating messages stemming from the creation of new records in the existing systems using the local terminology. In order to achieve this, SNOMED CT-AU or AMT reference sets can be used as a basis for mapping the local terminology codes to SNOMED CT-AU or the AMT, and vice versa.

In order to send a message, the local codes need to be mapped to a SNOMED CT-AU or AMT code. Where an agreed set of SNOMED CT-AU or AMT codes is being used in specific messages, these may be published as a messaging reference set. To receive a message, all of the messaging reference set data needs to be mapped to the local code sets, which may be a many-to-one or one-to-many mapping. However, if some SNOMED CT-AU or AMT terms in the reference sets are not applicable to the local system, then it could be mapped to an "error/human intervention required" local code.

The suggested approach is to flatten the terminology distribution files. For each reference set, a file needs to be created with the following fields:

- ConceptId
- Description text of the Preferred Term

A SQL query could be created to achieve this by joining the Descriptions table with each of the reference set files and the applicable Language reference set. The ADRS indicates the general language preferences for Australian implementations.

The resulting file would have sufficient information to generate two simple maps for each reference set, one for inbound messages and one for outbound messages. Once mapped, the file would now contain:

- ConceptId
- Reference set id
- Description text of the Preferred Term
- Local terminology code
- Local terminology text
- Effective date

When new versions of the messaging reference set files are released, the inbound message map must be updated. Hence, the recommendation is to include the

"Effective date" field in the mapped file to provide traceable and reproducible message translations. Also the maps can be updated by simply adding new rows to the table storing the mapped data.

This approach enables systems to continue to operate without large modifications, and without a change to the current user experience, while enabling the use of Australian terminology for information exchange and data reporting/analysis. Additionally the mapping method introduces features provided by the local or proprietary systems not provided natively within SNOMED CT-AU or the AMT.

However, maps can be expensive to produce and maintain over time, particularly if both the source and target products are frequently updated. Consideration should be given to the release and update cycles of both the local and Australian terminologies.

Additional guidance on terminology adoption via mapping and requirements on clinical messaging can be found in the following documents:

- *Clinical Terminology – Guidance for People and Processes* [8]
- *Clinical Terminology – Guidance for Use in Healthcare Software* [9]
- *Clinical Terminology – Use of Medical Nomenclatures in Information Exchange* [10]
- *AMT Mapping Guidelines* [11]

7.2 Native implementation

An alternative to mapping Australian terminology to a local or proprietary code set is to directly or natively implement the Australian terminology.

One of the first technical considerations when implementing the AMT or SNOMED CT-AU is storage and retrieval of terminology content as reference data. The aims of storing terminology as reference data are:

- to enable searching for values when entering transactional data;
- to render transactional data containing fields encoded with terminology; and
- to report across transactional data using the terminology reference data to group and filter.

There are two broad approaches that can be taken in this respect:

- custom schema and application code for the implementation; or
- use of an external terminology server.

Both have advantages and disadvantages that must be weighed when planning an implementation. Section 7.3 provides more detail.

Additional guidance on adopting the AMT natively can be found in the following documents:

- *Clinical Terminology – Guidance for People and Processes* [8].
- *Clinical Terminology – Guidance for Use in Healthcare Software* [9].

7.2.1 Limited native implementation

Scenario: Using Australian terminology as an interface terminology.

A limited native implementation may include the use of SNOMED CT-AU or AMT reference sets simply as an interface terminology. These may be as simple as drop-down or pick lists.

The implementer needs to undertake an analysis to determine the required list of concepts from the reference set to be displayed. Once this is determined, new reference sets can be created or existing ones modified to create subsets for different contexts and different user groups of the system.

The required description or display text for each concept must also be determined. In most instances, it is strongly recommended that the Preferred Term be used. However, other acceptable synonyms may also be used to display text.

Section 8.2 provides more detail on the technical considerations for recording, storing and display of terminology-encoded information.

Systems implementing SNOMED CT-AU or the AMT in this way are limited in that they do not take advantage of the information structures provided by the SNOMED CT or AMT concept models, and simply use the terminology as a standardised interface vocabulary.

7.2.2 Comprehensive native implementation

Scenario: Development of a new system using Australian terminology.

There is an opportunity to use SNOMED CT-AU or the AMT as the native coding system when new systems are developed, enabling easier interoperability. To implement SNOMED CT-AU or the AMT in a new system, it will be necessary to refer to both this document and the *SNOMED TIG*. New systems may choose to implement SNOMED CT-AU or AMT descriptions within the user interface, or provide a customised vocabulary mapped to underlying SNOMED CT codes or expressions for storage and transmission. Implementers of new systems are encouraged to incorporate the entire terminology release and may take advantage of the terminology structures to implement decision support.

7.3 Terminology server versus native implementation

Bespoke terminology data structures and implementation can provide fast functionality, well targeted to the system use cases. However, this approach does have disadvantages:

- More complexity in the application which might otherwise be outsourced to an external system (terminology server).
- Distraction for application developers from the main purpose of the system.
- More code to maintain and test.
- Design must take into account regular terminology updates.
- Reduced ability to centrally manage terminology for multiple systems in an organisation.
- Complex reasoning features provided by terminology servers are hard to reproduce cheaply.

Terminology servers provide an alternative which, depending upon the product chosen, can eliminate one or more of the above disadvantages. We recommend considering the following characteristics when assessing terminology servers:

External point of failure An external terminology server, depending upon the nature of the deployment, how it is used within an application (that is, real-time access or offline with updates), may present an additional point of failure within a deployment.

Third party component Introduction of a terminology server will include an additional third party component, requiring the usual licensing and contractual agreements.

Unless the implementation of terminology within an application is trivial, a terminology server is likely to be simpler and cheaper than implementing, testing and maintaining custom functionality. However, this decision must be made on a case-by-case basis.

8 Implementation considerations

8.1 Term searching and capturing input

The most common method of providing end users access to the broad range of content in SNOMED CT-AU and the AMT is to allow searches on appropriate concepts by entering key words and selecting a suitable description from the search results. To ensure a positive user experience, it is important to provide an effective search functionality. Section 7.6.1 "Text Searches" of the *SNOMED TIG* [1] provides some options for producing effective searches, and some additional approaches are described below.

8.1.1 Indexing

Basic indexing of the **term** column in the Descriptions file has limitations. Most users will expect to be able to retrieve search results without typing in the whole phrase they are looking for, which may even mean partial words. Such search criteria may require the use of wild cards.

For example, a user searching for "fracture" may just type in "frac".

The following query produces over 1,800 results with an acceptable performance:

```
SELECT term
FROM descriptions_snapshot
WHERE term LIKE 'frac%';
```

However, such "starts with" searches will not identify terms where "frac" appears other than at the start of the description. It is possible to prefix the search string with a wild card such that:

```
SELECT term
FROM descriptions_snapshot
WHERE term LIKE '%frac%';
```

This query will yield over 12,000 results that contain "frac" anywhere within the string. However, since basic SQL indexing typically relies on the characters at the front of a string, the index is useless and performance will suffer.

There are a number of alternatives that are both simple to use and provide good results. These include:

- **Native full text indexing:** Most relational database management systems (RDBMSs) now provide native full text indexing. The features and syntax vary between systems, so the relevant RDBMS user documentation should be consulted. Such indexes require little extra setting up and are easy to use.
- **Specialised software libraries**: There are a number of software libraries available that can also be used to produce powerful indexes and searching capability. Configuration will depend on the library chosen. Lucene[8] is an

[8] http://lucene.apache.org/

example of such a library; versions are available for most major development languages.

8.1.2 Restricting scope to reference sets

An unrestricted search on all of SNOMED CT-AU or the AMT will yield what appear to be duplicates to an end user, and risk the possible selection of a concept that is inappropriate for the context.

For example, a search for "ulcer" might retrieve the following SNOMED CT-AU concepts:

- 56208002 |*Ulcer*|
- 429040005 |*Ulcer*|

Each of these concepts represents a different idea. The first refers to a morphological abnormality, that is, an ulcer. The second is the actual disorder as might be recorded in a clinical encounter. Refer to the *SNOMED CT Editorial Guide* [12] for further details.

Similarly, some AMT concepts have identical terms (Preferred Term strings) but are distinct concepts such as:

- 21433011000036107 |*paracetamol*|
- 2442011000036104 |*paracetamol*|

The first is a *Medicinal product* concept and the second is an AMT *Substance* concept.

One way to restrict the scope is to limit the search to concepts from one or a set of reference sets of appropriate scope. For example, if searching for a value for a diagnosis field, only *Clinical finding* concepts might be appropriate, and the *Clinical finding foundation reference set* may be applied. For example, when searching for a value for a generic medicines product field that optionally can include a dose form and a strength (to generically prescribe a medicine), AMT *Medicinal product* and *Medicinal product unit of use* concepts may be appropriate. In this case, the AMT *Medicinal product reference set* and *Medicinal product unit of use reference set* should be applied to constrain the search results.

Similarly, certain data elements in NCTIS information models have value domains that are restricted to particular reference sets. Implementers must ensure that users only populate these fields using the relevant concepts. [3]

8.1.2.1 SQL examples

Consider a basic search for the term "ulcer" as described above.

Unconstrained, the search may be executed as:

```
SELECT conceptId, term
FROM descriptions_snapshot \
WHERE term LIKE 'ulcer';
```

The query should yield two results. To restrict the search to *Clinical findings*, use the reference set 32570071000036102 |*Clinical finding foundation reference set*|, as follows (assuming the reference set has been imported):

```
SELECT term
FROM descriptions_snapshot
WHERE term LIKE 'ulcer'
AND conceptId IN (SELECT referencedcomponentId
                  FROM refset_snapshot
                  WHERE refsetId = 32570071000036102);
```

Only the disorder concept 429040005 |*Ulcer*| is now returned. Note that even concepts within the same hierarchy may have identical Synonyms. Problems relating to this may be addressed by restricting results to Preferred Terms, as described in Section 8.1.3 below.

Developers may find it useful to create a function to determine if a given conceptId is a member of a certain reference set.

8.1.3 Use all synonyms but limit results to Preferred Terms

The availability of Synonyms provides a way for users to search for concepts using a variety of terms. However, Synonyms are not necessarily unique, even within a single hierarchy. So care must be taken to ensure that users select the concept that unambiguously matches their intentions. A search for "Period pain" may return two concepts with that Synonym, both from the *Clinical findings* hierarchy:

- 266599000 |*Period pain*|
- 289900009 |*Period pain*|

Additionally, a search may yield matches on several similar Synonyms for the same concept. A search of disorders containing the word "hypertension" might return the following terms:

- Hypertension
- HTN – Hypertension
- Systemic arterial hypertension
- HT – Hypertension
- BP+ – Hypertension

These terms are all synonyms for the same concept (38341003 |*Hypertensive disorder, systemic arterial (disorder)*|). However, from a user perspective, this can raise confusion or doubt as to which term they should choose. Different results might suggest different concepts.

One approach is to use all the available synonyms to execute the search and identify the associated concepts (using the conceptId), rather than displaying only the Preferred Terms for those concepts.

8.1.3.1 SQL examples

The simplest way to produce a list showing only individual concepts and Preferred Terms, is to create a function that extracts the Preferred Term for a particular concept from the *Australian dialect reference set*.

Reusing the example from before, searching for "Period pain" would result in unconstrained queries that might look like the following sample:

```
SELECT conceptId, term
FROM concepts_snapshot AS C
INNER JOIN descriptions_snapshot AS D
ON C.id = D.conceptid
WHERE term = 'Period pain'
AND C.active = 1
AND D.active = 1;
```

This returns the following:

- 266599000 |*Period pain*|
- 289900009 |*Period pain*|

We can change the query so that the result set provides the Preferred Terms, by using the precreated function which appears in bold in the query below:

```
SELECT conceptId, get_PT(conceptId)
FROM concepts_snapshot AS C
INNER JOIN descriptions_snapshot AS D
ON C.id = D.conceptid
WHERE term = 'Period pain'
AND C.active = 1
AND D.active = 1;
```

This revised query will now return the following results:

- 266599000 |*Dysmenorrhoea*|
- 289900009 |*Period pain present*|

Note that the same concepts are returned, but they can now be distinguished by an end user (who should generally not be exposed to the identifiers).

The same approach can also address the issue of multiple synonyms for the same concept being returned, by using the **DISTINCT** keyword.

```
SELECT DISTINCT(conceptId), get_PT(conceptId)
FROM concepts_snapshot AS C
INNER JOIN descriptions_snapshot AS D
ON C.id = D.conceptid
WHERE term LIKE '%hypertension%'
AND C.active = 1;
```

This query returns approximately 250 unique concepts, based on over 600 matching terms[9], including results like 398254007 |*Pre-eclampsia*|, which do not have the search phrase in the Preferred Term.

This approach leverages the synonyms in SNOMED CT-AU so that users can find concepts using phrases that are not necessarily the Preferred Terms. Although limiting the search results only to Preferred Terms should reduce the risks of too many choices, including misunderstanding, some users may not be comfortable with choosing a term that looks markedly different from their input. Developers need to consider their customers and provide an appropriate solution.

[9] Based on SNOMED CT-AU November 2013 data. Different releases may produce different results.

8.2 Recording, storage and display of clinical information

When a health software system displays terminology descriptions in the interface to the user for recording and storing clinical information, it is recommended that it store the following pieces of information:

The terminology concept identifier of the code selected. For example:

- If using SNOMED CT-AU natively, this could be the SNOMED CT identifier of an AMT concept like 21433011000036107.
- If using a mapped implementation, this will be the code from the local medicines dictionary for the selected item.

The preferred terminology description text seen and selected by the user. For example:

- If using SNOMED CT-AU natively, this could be the Preferred Term of the above AMT concept, |*paracetamol*|.
- If using a mapped implementation, this will be the description text of the local medicines dictionary that was displayed to the user.

The terminology release version being used at the time the clinical record was created.

The code and text are both stored to ensure that if any uncertainty arises, the stored text seen by the user is the definitive record. Additionally storing the release version could serve as an audit trail, to be potentially used for troubleshooting issues in message exchange and medico-legal requirements.

8.3 Identifying versions of terminology releases

When storing and using terminology component identifiers (for example, in clinical documents, maps, or terminology servers) the following URI string should be used to identify the version of the release:

```
"http://snomed.info/sct/{moduleId}/version/{effectiveTime}"
```

The effectiveTime value is the release date specified for the terminology release files. For example, a component released in the January 2016 release will have an effectiveTime value of "20160131".

Before November 2015, the AMT was published as a stand-alone terminology and all components used the AMT module identifier of 900062011000036108. An example of the string used to identify a component released in the AMT October 2015 release would be as follows:

```
"http://snomed.info/sct/900062011000036108/version/20151031"
```

From November 2015 the AMT is included as a formal subset of the SNOMED CT-AU release in preparation for future integration work, and to better support of the usage of terminology within the My Health Record. The combined release is provided in a single set of RF2 files and versioned using the SNOMED CT-AU module identifier 32506021000036107.

The resulting string for a component released in the November 2015 release becomes:

```
“http://snomed.info/sct/32506021000036107/version/20151130”
```

Implementations using just the AMT should use the same URI, as they are effectively using a subset of SNOMED CT-AU.

Examples of different encodings of the version of the SNOMED CT-AU January 2016 release are displayed below.

8.3.1 HL7 Clinical Document Architecture (CDA)

In a CDA document, the version of this release may be encoded in a Concept Descriptor field named “xyz” using the **codeSystemVersion** attribute as follows:

```
<xyz code="33256011000036105"
codeSystem="2.16.840.1.113883.6.96"
codeSystemName="SNOMED CT-AU"
codeSystemVersion="http://snomed.info/sct/32506021000036107/
version/20160131"
displayName="Lorano 10 mg tablet: uncoated, 30”/>
```

8.3.2 HL7™ Fast Healthcare Interoperability Resources (FHIR®)

In FHIR® resources, the version of this release may be encoded in a Coding field named “xyz” using the **version** element as follows:

XML example:

```
<xyz>
     <system value="http://snomed.info/sct" />
     <version
value="http://snomed.info/sct/32506021000036107/version/20160131" />
     <code value="33256011000036105" />
     <display value="Lorano 10 mg tablet: uncoated, 30” />
</xyz>
```

8.3.3 JSON

```
"xyz": [
   {
      "system": "http://snomed.info/sct",
      "version": "http://snomed.info/sct/32506021000036107/version/20160131",
      "code": "33256011000036105",
      "display": "Lorano 10 mg tablet: uncoated, 30"
   }
]
```

For further information on the URI standard that governs the application of this versioning, see the *SNOMED CT URI Standard* [13].

8.4 AMT-specific considerations

8.4.1 CTPP versus TPP

The use cases for an implementation need to be considered when choosing to use CTPP or TPP concepts, or both.

Where references to trade packs are required without specifying the container type, TPP concepts should be used. For instance, pack-based prescribing usually requires a product pack to be specified. However, it is not necessary to specify a container. Presenting users with a variety of container-based variations of a pack (bottle, blister pack) may frustrate users with irrelevant options and slow down data entry unnecessarily.

Under some circumstances, clinicians may however need to specify particular containers when prescribing. Similarly, the more specific CTPP concept is required when recording a specific dispensed medication. For example, when recording a medication dispensed from a community pharmacy via barcode scanning.

Therefore it is necessary to analyse system requirements before choosing when CTPP concepts, TPP concepts, or both are appropriate.

8.4.2 Description term length

There is a maximum field length of 2,048 characters specified for the term field in SNOMED CT-AU, to cater for the longer descriptions required to describe AMT concepts. However, in practice the current longest AMT terms are just over 1,000 characters. Depending upon the section of the AMT content being used, the longest terms actually present in the AMT may be significantly shorter than this limit.

For the AMT 20140630 release, 95% of active Preferred Terms were less than 93 characters.

The graph below shows the distribution of active Preferred Term lengths in the AMT June 2014 Production release.

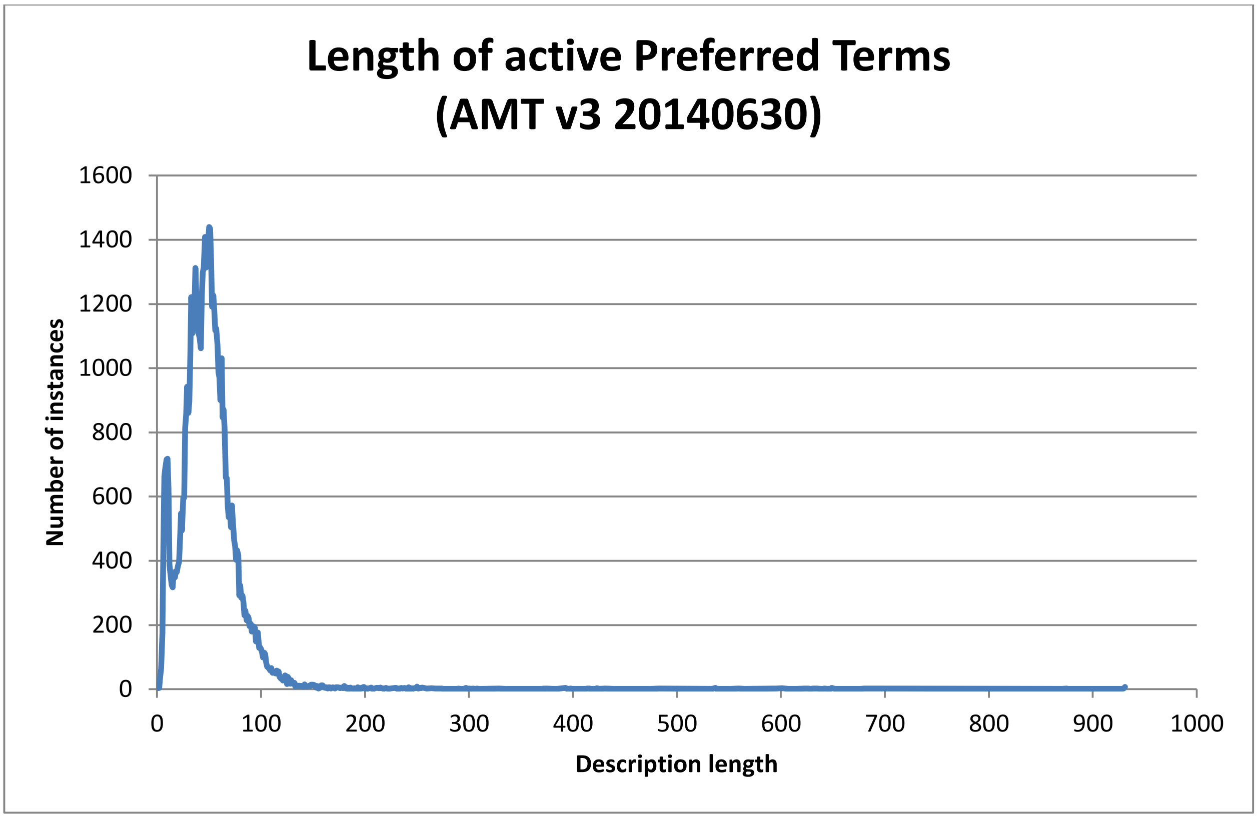


Figure 16: Active Preferred Term lengths in the AMT June 2014 production release

System developers needing to allocate less than 2,048 characters are advised to check the release files for the maximum description length for the AMT content they use on a per-release basis. This will help ensure that descriptions are not truncated, as this is clinically unsafe.

Any field length restriction for an application that results in truncation of AMT descriptions should lead to a discussion with the NCTIS for implementation guidance and clinical safety considerations.

Assessment and potential development of clinical interface descriptions for AMT concepts is included in the *AMT Survey Results and Roadmap* [14], which will also consider label names, sort order and ingredient order.

8.4.3 AMT and PBS data

As of December 2012, PBS data includes AMT concept identifiers and Preferred Terms for MP, MPUU, MPP, TPUU and TPP. The Pharmaceutical Benefits Division (PBD) is planning to increase their AMT adoption, which will eventually encompass other the AMT concepts beyond those mentioned.

And more specifically, the AMT MPUU and TPUU concepts are used to represent chemotherapy items as included in the PBS.

Where an AMT concept is not available or does not meet PBS needs, a non-AMT PBS identifier is generated. These concept identifiers include a PBS-specific SNOMED CT namespace identifier of "1000144", as seen in the identifier 57291000144108. These concepts are published within the PBS data outputs and are maintained by the PBD.

For further information on the PBS implementation of AMT data, see the NEHTA AMT-PBS FAQs[10] and *Pharmaceutical Benefits Scheme (PBS)* [15].

8.4.4 Strength reference set considerations

8.4.4.1 Sufficiency of floating point strength accuracy

When normalising values to populate the value field of the *Strength reference set*, the result includes some real numbers that repeat after the decimal point (for example, 0.333333333...). It was determined that up to 13 significant figures are required to arrive at a sufficiently accurate strength value when calculating the non-normalised strength attribute, such as 16666.66666667, 33333.33333333 and 149.25373134.

For example the strength value of the MPUU *epoetin beta 5000 international units/0.3 mL injection, syringe* is 16666.66666667. This product has a Unit of use size of 0.3 mL, thus recalculating non-normalised (human readable) strength results in "5000.000000001 international units/0.3 mL" which affords sufficient accuracy.

Note that if arithmetic is performed using AMT floating point strengths, issues relating to rounding of recurring numbers may be encountered and should be taken into account. The following table of MPUUs provides examples which can be used for testing.

Table 7: Sample MPUU floating point strength values

MPUU ID	MPUU Preferred Term	BoSS	Strength Value	Strength Unit
21996011000036108	epoetin beta 5000 international units/0.3 mL injection, syringe	epoetin beta	16666.66666667	international unit/mL
22082011000036102	follitropin beta 900 international units/1.08 mL injection, cartridge	follitropin beta	833.33333333	international unit/mL
23315011000036101	anakinra 100 mg/0.67 mL injection, syringe	anakinra	14925.373134	microgram/mL

[10] http://www.nehta.gov.au/our-work/clinical-terminology/australian-medicines-terminology/amt-pbs-faqs

8.4.4.2 Unit conversion

Unit conversion occurs in the *Strength reference set* when:

- Different MPUU concepts share the same BoSS, meaning that they have the same destination *Substance* for the HAS AUSTRALIAN BoSS relationship.
- Different strength units are used. For example, "microgram/mL" and "mg/mL".

A single unit of measure is chosen to represent the strength for this set of MPUUs (for example, only to "mg/mL"). This is to allow easier automated strength comparisons across similar products in the absence of machine-computable unit conversion factors.

This normalisation of units of measure does not affect MPUU (and other) descriptions, where they may represent the strength in the most useful form to humans that is equivalent to the machine representation in the *Strength reference set*.

The table below describes an example of unit conversion in the AMT *Strength reference set*.

Table 8: MPUU strengths with and without unit conversions

MPUU Preferred Term	**BoSS**	**Denormalised/ human readable strength (in MPUU Preferred Term)**	**Normalised strength (in *Strength reference set*)**	**Units converted**
metformin hydrochloride 500 mg tablet	metformin hydrochloride	500 mg	500 mg/each	No
metformin hydrochloride 850 mg tablet	metformin hydrochloride	850 mg	850 mg/each	No
metformin hydrochloride 1 g tablet	metformin hydrochloride	1 g	1000 mg/each	Yes

The last row in the table shows a unit conversion from "g" to "mg" along with an equivalent strength value conversion.

Implementers who are interested in using the atomically accessible strength details (calculated from the *Strength reference set* and the *Unit of use size reference set*) should consider if the calculated strength is appropriate for use when unit conversion has occurred. While the calculated strength is equivalent to the human readable strength in the MPUU description, it may not appear in identical characters (that is, not a lexical match).

Using the same example as above, the MPUU description's human readable strength is "1 g" while the calculated strength is "1000 mg". Unit conversion factors are not currently included in the AMT data.

8.4.4.3 Unit of measure of patches

Certain products with a *Form* of “patch” use the strength value and unit to represent a release rate in the *Strength reference set.* This is considered more clinically relevant than the total amount of an active substance. The release rate representation is not calculated to a denominator of one, which is typical for the strength value field.

For example, the MPUU |*testosterone 5 mg/24 hours patch*| has a strength value of “5” and a *Composite unit of measure* of |*mg/24 hours*|.

9 Configuration scripts and sample SQL Queries

This section outlines the content and purpose of the **NCTIS_Australian_Terminology_Sample_Scripts.zip** file.

The file contains configuration scripts for the set-up of a sample database schema into which the content of the terminology Release Bundle is loaded. This collection of SQL scripts is not a suggested approach for implementing the terminologies. Their purpose is to act as a learning tool to understand the release format and data structures of SNOMED CT-AU and the AMT in a relational database environment.

The following table identifies each file contained within the compressed script bundle along with a brief description of their use.

Table 9: Configuration scripts and queries for sample database schema

File Name	Script Description
schema/1_createSchema.sql	Creates the database schema and core tables.
schema/2_populateTables.sql	Script to load content from respective data files from the release bundle into tables created in the step above.
schema/3_createIndexes.sql	Creates indexes for the associated tables. These are generally added after loading the data, to maximise the performance of the import.
schema/4_createRoutines.sql	Creates functions to extract the Fully Specified Name and Preferred Term descriptions for a given concept.
schema/5_createTransitiveClosure.sql	Procedure to create a transitive closure table and associated indexes.
schema/6_createAMTObjects.sql	Creates AMT specific tables and views such as concrete domains and precoordinated tables.
schema/7_populateAMTObjects.sql	Script to load and insert data into objects created in the script above.
schema/8_createAMTIndexes.sql	Creates indexes for AMT tables created in step 6.
sql/AMT_use_cases.sql	Queries to extract data for AMT use cases.
sql/query_extract.sql	List of queries used throughout this document.
sql/sample_queries.sql	Additional example queries.
sql/seven_notable_concepts.sql	Lists out all the components of the reference set for the seven notable concepts in the AMT.

Note: The files listed in the table above need to be executed in the order in which they are listed. The step number is indicated in the file name.

9.1 Database schema definition

The sample database schema for SNOMED CT-AU and AMT content should consist of a minimum of six tables. The correspondences between terminological entities and their respective database tables are summarised below.

Table 10: Entity to table mapping

Entity	**Table Name**	**Details**
Concepts	**concepts_snapshot**	Covers clinical ideas and medicine products
Relationships	**relationships_snapshot**	Contains the associations between two concepts.
Descriptions	**descriptions_snapshot**	Contains the human readable terms for each concept.
Concept reference set	**refset_snapshot**	Contains the Clinical, Foundation and seven Notable concepts reference sets. Entries refer to a concept component.
Description reference set	**language_refset_snapshot**	Contains the *Australian dialect reference set*. The entries refer to a description component. An acceptability attribute indicates whether a description is the preferred or acceptable synonym.
Transitive closure	**transitive_closure**	Contains all the IS A relationships within SNOMED CT-AU and the AMT.

9.1.1 Entity-relationship diagram

The diagram below shows the schema design for the core SNOMED CT tables.

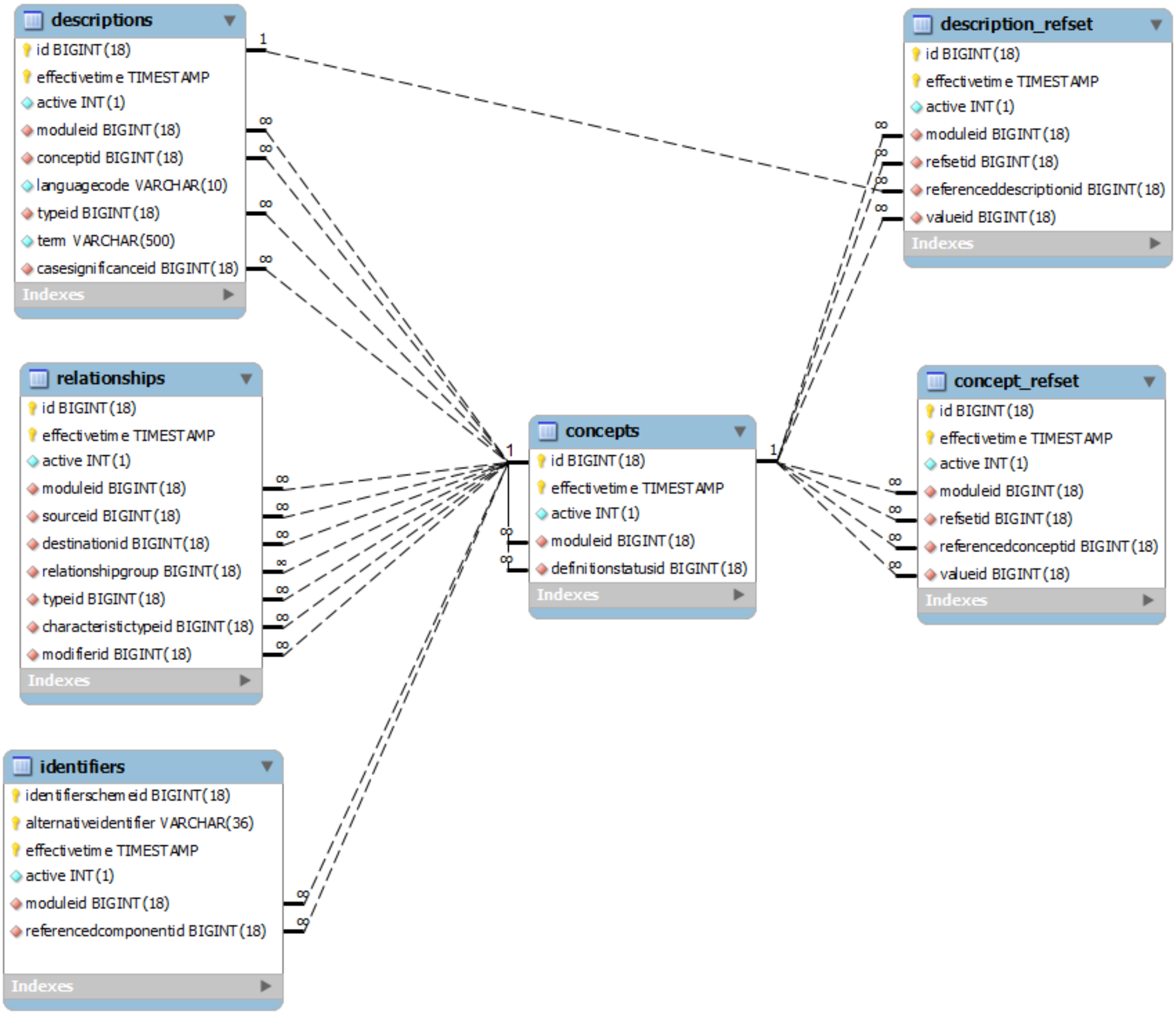


Figure 17: Schema design for core terminology tables

In the interest of providing an illustrative example of schema configuration and to gain an understanding of how the core components are linked to each other, it is sufficient to create the simplified set of tables as outlined in the diagram below.

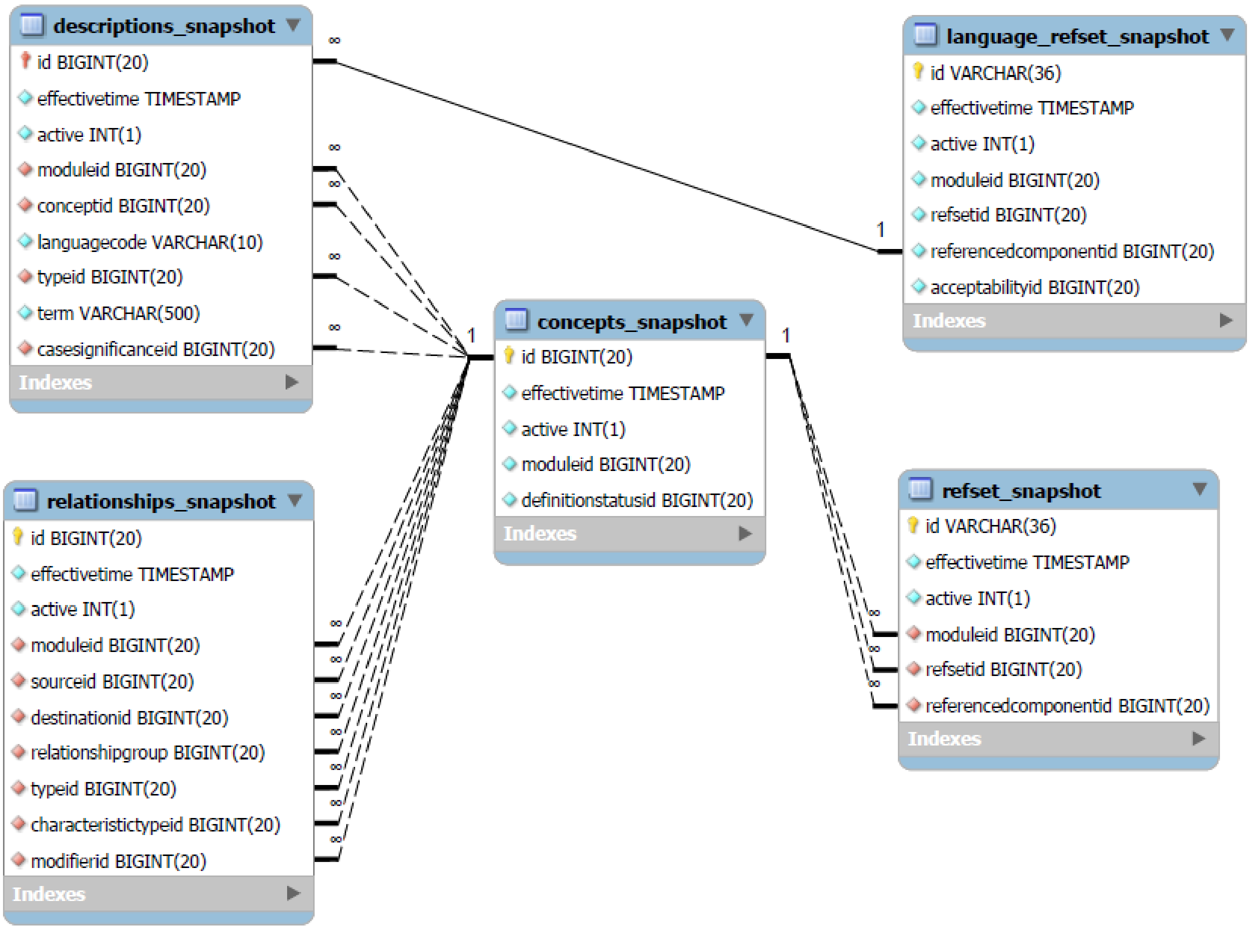


Figure 18: Entity-relationship diagram for sample schema

9.2 Database design notes

The following elements of design rationale in the database schema should be considered:

- **MySQL**

  The creation commands for tables, views, indexes and SQL queries[11] contained within the sample scripts file have been constructed within and tested against a MySQL database. This product was chosen as it is an open source relational database platform available as a free download.

- **No referential integrity**

  Foreign key constraints have been specified in the creation commands to define the relationships between tables in the schema. However, at the same time the use of the MyISAM database engine[12] has been specified. MyISAM will ignore these constraints and does not provide referential integrity checking or transaction support. As the sample SELECT queries do not require referential integrity, it was deemed acceptable to use MyISAM, which provides a significant performance advantage over other engines, particularly during the data load process.

[11] SQL = Structured Query Language.

[12] See http://dev.mysql.com/doc/refman/5.7/en/myisam-storage-engine.html

- **Snapshot release type**

  The sample scripts load data from the Snapshot release files, which contain the most recent version of each released component. However, if the complete history of each component is required in a production environment, then the Full release files can be used instead. Subsequently, Snapshot tables can be derived from the Full versions using the **effectiveTime** column to extract the current version of every component.

  ```
  CREATE TABLE concepts_snapshot AS
  SELECT t1.* FROM concepts_full t1
  WHERE t1.effectivetime = (
      SELECT MAX(t2.effectivetime)
      FROM concepts_full t2
      WHERE t1.id = t2.id);
  ```

  In the statement above, the **concepts_snapshot** table is created using a subquery that extracts the most recent version of every concept from a Full release.

9.3 Preparation for schema creation

This section assumes the reader has installed MySQL and created a database schema into which the SNOMED CT-AU release files can be loaded. If this is not the case, http://www.mysql.com provides free downloads of their MySQL Community Server. Additionally the installation and configuration instructions can be found on the download site.

The scripts assume that the SNOMED CT-AU Release File Bundle and the Australian Terminology Sample Scripts are extracted in the following directory structure:

```
<root-directory>/
   <release-files>/
      RF2Release/

Australian_Terminology_Sample_Scripts/
      schema/
         1_createSchema.sql
         2_populateTables.sql
         3_createIndexes.sql
         4_createRoutines.sql
         5_createTransitiveClosure.sql
         6_createAMTObjects.sql
         7_populateAMTObjects.sql
         8_createAMTIndexes.sql
      sql/
         AMT_use_cases.sql
         query_extract.sql
         sample_queries.sql
         seven_notable_concepts.sql
```

With a current working directory of <some-root-directory>, start a MySQL session.[13] The scripts listed above have a step number in the file name to indicate the sequence in which they need to be executed. Also the assumption is that the reader uses the command line version of the MySQL client, rather than MySQL Workbench.

Note:

- The above scripts have been saved in the default MySQL CLI format, using CR (carriage return) without LF (line feed). For this reason, these files will not display line endings correctly in some Windows (and older Mac) based programs. Please refrain from opening these files in MS Notepad and use an editor that honours CR line endings when displaying and writing the file content.
- The script **Australian_Terminology_Sample_Scripts/schema/ populateTables.sql** contains relative paths to the release files. Depending on the operating system and version of MySQL, please amend these and replace with the full path. For example, if using the terminology release bundle from January 2016:

  <release-files>/RF2Release/Snapshot/Terminology/ sct2_Concept_Full_AU1000036_20160131.txt

  changes to

  C:/Downloads/EP_2227_2016_ClinicalTerminology_v20160131/ NEHTA_2225_2016_SNOMEDCT-AU_CombinedReleaseFile_ v20160131/RF2Release/Snapshot/Terminology/sct2_Concept_ Full_AU1000036_20160131.txt

Once the SNOMED CT-AU and AMT data has been successfully imported, the content can be retrieved using the sample queries contained within the **Australian_Terminology_Sample_Scripts/sql** directory.

It should be noted that these queries are provided as a starting point to demonstrate a general means of importing and querying the terminology content. They are intended for purely illustrative purposes only, and are not appropriate for any other use.

9.4 SNOMED CT-AU sample queries

The queries discussed in this section are contained in the **Australian_Terminology_Sample_Scripts/sql/** directory.

The queries that are in the script bundle are based on scenarios that have been deemed to be generally useful.

[13] Instructions on how to open a mysql session and how to execute commands are available on the MySQL website http://www.mysql.com

9.4.1 Finding an active concept through a term or description

```
SELECT *
FROM descriptions_snapshot
WHERE term LIKE 'Myocardial infarction%'
AND active = 1
ORDER BY conceptid;
```

9.4.2 Retrieving the Fully Specified Name and acceptable Synonyms for a particular concept

```
SELECT
conceptid, term,
CASE typeid
   WHEN 900000000000013009
          THEN 'Synonym' ELSE 'Fully specified name' END
AS description_type
FROM descriptions_snapshot
WHERE term LIKE 'abdominal aorta finding%'
AND active=1;
```

9.4.3 Listing preferred descriptions of all active concepts in a particular reference set

This example is based on the *Adverse reaction type reference set*.

```
SELECT
c.id AS conceptid,
d.id AS descriptionid,
d.term AS preferred_term
FROM
concepts_snapshot AS c,
refset_snapshot AS rs,
descriptions_snapshot AS d,
language_refset_snapshot AS adrs
WHERE c.id=rs.referencedComponentId
AND c.id=d.conceptid
AND d.id=adrs.referencedComponentId
AND adrs.acceptabilityid=900000000000548007 -- ID of Preferred Term
AND rs.refsetid= 11000036103 -- ID of Adverse reaction type refset
AND c.active=1
AND d.active=1
AND rs.active=1
ORDER BY preferred_term;
```

9.4.4 Generating a list of all reference sets and the number of member concepts that belong to each reference set

```
SELECT
desc_active.term AS "Name of Reference Set",
refset_active.refsetid AS "Reference Set ID",
refset_active.member_count AS "No of Members"
FROM
       (SELECT
       term,id,conceptid
       FROM descriptions_snapshot AS ds
       WHERE active=1) AS desc_active,

       (SELECT
              referencedComponentId
              FROM language_refset_snapshot AS lrs
              WHERE refsetId = 32570271000036106 -- Australian dialect refset
              AND acceptabilityid = 900000000000548007  -- Preferred Term
              AND active=1) AS lang_refset_active,

              (SELECT
              refsetid, COUNT(referencedcomponentid) AS member_count
       FROM rf2_refset_snapshot AS rss
              WHERE active=1
       GROUP BY refsetid) AS refset_active

WHERE desc_active.conceptid = refset_active.refsetid
AND desc_active.id = lang_refset_active.referencedcomponentid
ORDER by desc_active.term;
```

9.4.5 Searching for descendants of a particular concept

This example is based on the concept 106112009 |*Fetal finding (finding)*|).

```
SELECT
c.id AS conceptid,
d.id AS descriptionid,
d.term AS preferred_term
FROM
concepts_snapshot AS c,

(SELECT sourceId
FROM transitive_closure
WHERE destinationId=106112009 -- Fetal finding
) AS ffd,

descriptions_snapshot AS d,
language_refset_snapshot AS adrs
WHERE c.id=ffd.sourceid
AND c.id=d.conceptid
AND d.id=adrs.referencedComponentId
AND adrs.acceptabilityid=900000000000548007 -- ID of Preferred Term
AND c.active=1
AND d.active=1
AND adrs.active=1;
```

9.4.6 Applying a grouper exclusion set

This example applies the *Clinical finding grouper exclusion reference set* against the *Fetal finding* hierarchy.

```
SELECT
c.id AS conceptid,
d.id AS descriptionid,
d.term AS preferred_term
FROM
concepts_snapshot AS c,

(SELECT sourceid
FROM transitive_closure
WHERE destinationid=106112009 -- Fetal finding
AND sourceid NOT IN
       (SELECT referencedcomponentid
       FROM refset_snapshot
       WHERE refsetid = 171991000036103 -- clinical finding grouper exclusion
refset
       AND active=1
       )) AS ffd,

descriptions_snapshot AS d,
language_refset_snapshot AS adrs
WHERE c.id=ffd.sourceid
AND c.id=d.conceptid
AND d.id=adrs.referencedComponentId
AND adrs.acceptabilityid=900000000000548007 -- ID of Preferred Term
AND c.active=1
AND d.active=1
AND adrs.active=1;
```

9.4.7 Finding terms within a specific hierarchy

```
SELECT
d.term AS preferred_term

FROM
concepts_snapshot AS c,

(SELECT sourceid
FROM transitive_closure
WHERE destinationid=71388002 -- Procedure hierarchy
) AS pd,

descriptions_snapshot AS d,
language_refset_snapshot AS adrs

WHERE
c.id = pd.sourceid
AND c.id=d.conceptid
AND d.id=adrs.referencedComponentId
AND adrs.acceptabilityid=900000000000548007 -- ID of Preferred Term
AND c.active=1
AND d.active=1
AND adrs.active=1
AND d.term like '% obstetric%';
```

9.5 AMT sample queries

As the AMT concept model differs from that of SNOMED CT-AU with a focus on medicinal products, additional queries have been developed to illustrate the specific features of the AMT.

The following sections look at how the sample tables and views can be used to extract data to fulfil the core AMT use cases – prescribing and dispensing.

The queries discussed in this section are contained in **sql/AMT_use_cases.sql**.

9.5.1 Queries to extract notable concepts

There are several valid approaches to developing a query which extracts every member of a notable concept class, like *Medicinal product pack*.

9.5.1.1 Using Notable concept reference sets

One method of obtaining all concepts within a notable class is to use the corresponding Notable concept reference set, included as part of the AMT content in the release file bundle. For example 929360081000036101 |*Medicinal product pack reference set*|.

Every child concept of a particular notable class is also given membership to its corresponding Notable concept reference set. The component referenced by the reference set member is the child concept. This can be queried as follows:

```
SELECT member.referencedcomponentid
FROM refset_snapshot AS member
WHERE member.refsetid = (
    SELECT conceptid
    FROM descriptions_snapshot
    WHERE term = 'medicinal product pack reference set (foundation metadata
concept)'
    AND active = 1)
AND member.active = 1
```

The above query uses the **refset_snapshot** table, which effectively contains data from all of the snapshot reference set files contained in the release. The **refsetId** column identifies the reference set (in this case, the *Medicinal product pack reference set*), and the **referencedComponentId** identifies the child concept of the notable class.

9.5.1.2 Using the transitive closure table

Another method is to find all source concepts that have an active IS A relationship to a destination concept of *Medicinal product pack*. The transitive closure table could be used to return all descendant rows as shown below:

```
SELECT sourceid
FROM transitive_closure
WHERE destinationid = 30513011000036104 -- medicinal product pack
AND sourceid NOT IN (
             SELECT sourceid
             FROM transitive_closure
             WHERE destinationid =  30404011000036106 -- trade product pack
);
```

9.5.2 Derived model

Queries based on the basic schema used so far tend to become verbose, repetitive and difficult to maintain when extracting anything but isolated pieces of data from the AMT model. For complex queries, it is often useful to create additional objects like views and query-based tables to provide fast and efficient access to the specific data required for these scenarios.

The following sections provide AMT-specific queries that could be used to satisfy the core uses cases of prescribing and dispensing. These queries focus on searching MPs, MPPs and TPPs. Some additional derived tables have been created to demonstrate this, and are discussed before delving into the use case data queries.

9.5.2.1 Unit of use

In order to extract the unit of use size and quantity for a given MPP, a query would need to navigate the MPP HAS MPUU relationship, the MPUU HAS UNIT OF USE relationship, the *Unit of use size reference set* and the *Unit of use quantity reference set*. By writing this query once to populate a derived table, the task of creating business queries to extract unit of use data is greatly simplified. An example of this query for an MPP is shown below:

```
SELECT
    MPPhasMPUU.sourceId AS mppid,
    get_PT(MPPhasMPUU.sourceId) AS mppterm,
    MPPhasMPUU.destinationId AS mpuuid,
    get_PT(MPPhasMPUU.destinationId) AS mpuuterm,

    hasUnitOfUse.destinationId AS unitofuseid,
    get_PT(hasUnitOfUse.destinationId) AS unitofuseterm,

    uouSize.operatorid AS sizeoperatorid,
    get_PT(uouSize.operatorid) AS sizeoperatorterm,
    uouSize.value AS sizevalue,
    uouSize.unitid AS sizeunitid,
    get_PT(uouSize.unitid) AS sizeunitterm,

    uouQty.operatorid AS quantityoperatorid,
    get_PT(uouQty.operatorid) AS quantityoperatorterm,
    uouQty.value AS quantityvalue,
    uouQty.unitid AS quantityunitid,
    get_PT (uouQty.unitid) AS quantityunitterm

FROM relationships_snapshot AS MPPhasMPUU

LEFT OUTER JOIN relationships_snapshot AS hasUnitOfUse
     ON MPPhasMPUU.destinationId = hasUnitOfUse.sourceId
     AND MPPhasMPUU.sourceId IN (SELECT id FROM amt_mpp)
     AND MPPhasMPUU.destinationId IN (SELECT id FROM amt_mpuu)
     AND hasUnitOfUse.typeId = 30548011000036101 -- has unit of use
(relationship type)
     AND hasUnitOfUse.active = 1

JOIN amt_uou_size_rs AS uouSize
     ON hasUnitOfUse.id = uouSize.referencedcomponentid
     AND uouSize.active = 1

JOIN amt_uou_qty_rs AS uouQty
     ON MPPhasMPUU.id = uouQty.referencedcomponentid
     AND uouQty.active = 1

WHERE MPPhasMPUU.typeId = 30348011000036104

AND MPPhasMPUU.sourceId = 26535011000036103 -- ethinyloestradiol 35 microgram +
norethisterone 1 mg tablet, 84 [4 x 21]

AND MPPhasMPUU.active = 1 -- ethinyloestradiol 35 microgram + norethisterone 1 mg
tablet, 84 [4 x 21]
;
```

Examining this in a little more detail, for a known MPP concept (26535011000036103 |*norethisterone 1 mg + ethinyloestradiol 35 microgram tablet, 4 x 21*|), the query:

1. Finds all HAS MPUU relationships for that MPP.
2. For each MPUU identified, the query joins to the **relationships_snapshot** table to find the HAS UNIT OF USE relationship for that MPUU.
3. The query joins to the **amt_uou_size_refset** table to find the reference set member which relates to the HAS UNIT OF USE relationship above.

The insert-select statement which queries the basic tables to populate the **amt_unit_of_use** table is based on the query structure described above. The query can be simplified using that table as follows:

```
SELECT *
FROM amt_unit_of_use
WHERE mppid = 26535011000036103;
```

9.5.2.2 Ingredient strength

In order to extract ingredients and their respective strengths for a given MPP, a query would need to navigate the MPP HAS MPUU relationship, the MPUU HAS INTENDED ACTIVE INGREDIENT relationship, the MPUU HAS AUSTRALIAN BoSS relationship, and finally the *Strength reference set*. By writing this query once to populate a derived table, the task of creating business queries to extract ingredient strengths is greatly simplified. An example of this query is shown below:

```
SELECT
    MPPhasMPUU.sourceId AS mppid,
    get_PT(MPPhasMPUU.sourceId) AS mppterm,
    MPPhasMPUU.destinationid AS mpuuid,
    get_PT(MPPhasMPUU.destinationid) AS mpuuterm,
    hasIngredient.destinationid AS substanceid,
    get_PT(hasIngredient.destinationid) AS substanceterm,
    hasBoSS.destinationid AS bossid,
    get_PT(hasBoSS.destinationid) AS bossterm,
    strength.operatorid AS operatorid,
    get_PT(strength.operatorid) AS operatorterm,
    strength.value AS strengthvalue,
    strength.unitid AS unitid,
    get_PT(strength.unitid) AS unitterm
FROM relationships_snapshot AS MPPhasMPUU

JOIN relationships_snapshot AS hasIngredient
     ON MPPhasMPUU.destinationId = hasIngredient.sourceId
     AND MPPhasMPUU.sourceId in (SELECT id FROM amt_mpp)
     AND MPPhasMPUU.destinationId in (SELECT id FROM amt_mpuu)
     AND MPPhasMPUU.typeId = 30348011000036104 -- has MPUU (relationship
type)
     AND MPPhasMPUU.active = 1
     AND hasIngredient.typeId = 700000081000036101 -- has intended active
ingredient (attribute)
     AND hasIngredient.active = 1

JOIN relationships_snapshot AS hasBoSS
     ON hasIngredient.sourceId = hasBoSS.sourceId
     AND hasIngredient.relationshipgroup = hasBoSS.relationshipgroup
     AND hasBoSS.typeId = 30364011000036101 -- has Australian BoSS
(relationship type)
     AND hasBoSS.active = 1 ;

LEFT OUTER JOIN amt_strength_refset AS strength
     ON hasBoSS.id = strength.referencedcomponentid
     AND strength.active = 1
WHERE MPPhasMPUU.sourceId = 26535011000036103 --  ethinyloestradiol 35
microgram + norethisterone 1 mg tablet, 84 [4 x 21]
;
```

Note that the strengths returned by this query are normalised to a denominator of one, as expressed in the *Strength reference set*.

Examining this in a little more detail, for a known MPP concept (26535011000036103 |*norethisterone 1 mg + ethinyloestradiol 35 microgram tablet, 4 x 21*|), the query:

1. Joins to the **relationships_snapshot** table to find any HAS MPUU relationships for that MPP.
2. For each MPUU identified (as the destination of the HAS MPUU relationship), the query joins to the **relationships_snapshot** table to find any HAS INTENDED ACTIVE INGREDIENT relationships for that MPUU.
3. The query then joins to the **relationships_snapshot** table to find the HAS AUSTRALIAN BoSS relationship from the same MPUU with the same **relationshipgroup** as the HAS INTENDED ACTIVE INGREDIENT relationship above.

   Note: the BoSS is the substance used to express the strength of an active MPUU ingredient, which may differ from the active component of the substance. For example:

   - HAS INTENDED ACTIVE INGREDIENT = Dexamphetamine (base)
   - HAS AUSTRALIAN BoSS = Dexamphetamine sulfate (salt)
4. Finally, the query joins to the **amt_strength_refset** table to find the *Strength reference set* member that relates to the HAS AUSTRALIAN BoSS relationship above.

The insert-select statement, which queries the basic table to populate the **amt_ingredient_strength** table is based on the query structure described above, and allows that query to be rewritten as:

```
SELECT *
FROM amt_ingredient_strength
WHERE mppid = 26535011000036103;
```

9.5.2.3 Combining strength and unit of use size

A query could be used to extract the total quantity of each ingredient contained in an MPUU or to derive the human friendly, denormalised strength as displayed in MPUU terms.

This query combines the *Strength reference set* with the *Unit of use size reference set*, and may be used to determine:

- The total quantity of an ingredient in the MPUU – for example if the strength is 15 mg/mL and the unit of use size is 20 mL, then the total ingredient quantity in the MPUU is 300 mg.
- The denormalised strength as represented in the MPUU descriptions, which are always based on the unit of use size. For example if the strength is 15 mg/mL and the unit of use size is 20 mL, then the strength represented in the MPUU description will be 300 mg/20 mL.

In order to extract that data, a query would need to navigate:

- the MPUU HAS AUSTRALIAN BoSS relationship;

- the MPUU HAS UNIT OF USE relationship;
- the *Strength reference set*;
- the *Unit of use size reference set*; and
- the *Composite unit of measure* relationships HAS NUMERATOR UNITS and HAS DENOMINATOR UNITS.

By writing this query once to populate a derived schema table, the task of creating business queries to extract ingredient strengths is greatly simplified. An example of this query is shown below:

```
SELECT
    strength.mpuuid AS mpuuid,
    strength.mpuuterm AS mpuuterm,
    strength.bossid AS bossid,
    strength.bossterm AS bossterm,
    strength.strengthvalue AS strengthvalue,
    strength.unitid AS strengthunitid,
    unitterm AS strengthunitterm,
    substanceid AS activeingredientid,
    substanceterm AS activeingredientterm,
    sizevalue AS sizevalue,
    sizeunitid AS sizeunitid,
    sizeunitterm AS sizeunitterm,
    ROUND(strength.strengthvalue * sizevalue, 6) AS totalquantity,
    hasNumeratorUnits.destinationid AS totalquantityunitid,
    get_PT(hasNumeratorUnits.destinationid) AS totalquantityunitterm
FROM amt_ingredient_strength AS strength

JOIN amt_unit_of_use AS uousize
     ON strength.mpuuid = uousize.mpuuid

JOIN relationships_snapshot AS hasNumeratorUnits
     ON strength.unitid = sourceId
     AND typeid = 700000091000036104
     AND active = 1
WHERE strength.mpuuid = 22148011000036103;
```

Examining this in a little more detail, for a known MPUU conceptId (22148011000036103[14]), the query:

1. Identifies the ingredients and strength using the derived **amt_ingredient_strength** table.
2. Joins to the Snapshot-derived **amt_unit_of_use** table to find the HAS UNIT OF USE relationship for that MPUU, along with the associated *Unit of use size* and *Unit of use quantity*.
3. Joins to the **relationships_snapshot** table to find the HAS NUMERATOR UNITS relationship for the *Composite unit of measure* concept identified by the *Strength reference set* member above.
4. Multiplies the strength value by the *Unit of use size* value to compute the total ingredient quantity (rounded to six decimal places).

[14] The full concept is 22148011000036103 |*polygeline 17.5 g/500 mL + potassium 99.71 mg/500 mL + sodium 1.67 g/500 mL + calcium 125 mg/500 mL + chloride 2.574 g/500 mL injection, bottle*|.

Both the HAS AUSTRALIAN BoSS and HAS INTENDED ACTIVE INGREDIENT relationships are required as the strength of the intended active ingredient is expressed in terms of the BoSS. That is:

- the HAS INTENDED ACTIVE INGREDIENT identifies the intended ingredient in the medication; and
- the HAS AUSTRALIAN BoSS identifies the substance that the strength of the ingredient is expressed in terms of.

The insert-select statement which queries the basic table to populate the **amt_ingredient_strength** table is based on the query structure described above, and allows that query to be rewritten as:

```
SELECT *
FROM amt_total_ingredient_qty
WHERE mpuuid = 22148011000036103;
```

Note: There is some inconsistency in the AMT data with respect to the way that unit of use size has been modelled, which means that for some MPUU concepts the total ingredient quantity cannot be reliably calculated. This affects some injections and powders.
As an example, consider the MPUU 22420011000036103 |*sodium bicarbonate 1.76 g + citrate sodium anhydrous 630 mg + citric acid 720 mg + tartaric acid 890 mg oral liquid: powder for, 4 g sachet*|.
The *Unit of use size* of this example is represented as "4 g", whereas it should be "1 sachet". As a result, the total ingredient quantity calculated for this MPUU for ingredient citric acid is actually 2880 mg, instead of 720 mg. This issue has not been resolved at the time of writing.

9.5.2.4 Query for prescribing use case

The goal of the Prescribing use case in the *AMT v3 Overview and Detailed Business Use Cases* [12] is stated as being:

> *"to support using the AMT as the source of the medicines terminology in the prescription of pack-based prescribing by an authorised prescriber, such as a General Practitioner (GP), and to support the generation and exchange of such information in a community-based model involving an authorised dispenser (e.g. a community pharmacy)".*

For simplicity, the derived model used in the sample scripts primarily focuses on prescribing by MPP and TPP concepts only. The product pack level has been chosen as it strikes a good balance between illustrating the product components, within the context of the AMT model, and targeting a level of refinement that is most intuitively prescribable. Where prescription by CTPP, MPUU or TPUU is required, these samples can be easily adapted via the relationships from, and between, MPP, TPP and CTPP.

In the following example, the prescriber has decided to prescribe amoxycillin, and has entered the characters "amox" into the medications search field of their prescribing system. While this is a simplified example, a real world implementation would likely dynamically query and refine these results as the user types each character. Further consideration to performance would also be required. For the intent of this example a simple transactional text search is illustrated.

The system will display a list of MPPs and TPPs for which their Preferred Term either contains a word commencing with the characters “amox”, or they contain a substance whose Preferred Term commences with the characters “amox”. The majority of the task can be achieved by simply querying the derived tables **amt_mpp_to_tpp** and **amt_ingredient_strength** as follows:

```
SELECT
    amt_mpp_to_tpp.mppid,
    amt_mpp_to_tpp.mppterm,
    amt_mpp_to_tpp.tppid,
    amt_mpp_to_tpp.tppterm
FROM amt_mpp_to_tpp

JOIN amt_ingredient_strength
     ON amt_mpp_to_tpp.mppid = amt_ingredient_strength.mppid

WHERE amt_ingredient_strength.substanceterm REGEXP (@search_term:='(^|[^a-zA-
Z]+)amox' collate utf8_unicode_ci)

OR amt_mpp_to_tpp.mppterm REGEXP @search_term
OR amt_mpp_to_tpp.tppterm LIKE @search_term
;
```

The above query primarily searches the **amt_ingredient_strength** table as it is already populated with terms for the MPPs, MPUUs and substances within them. We additionally join to the **amt_mpp_to_tpp** table to provide the ability to search the Preferred Term of the TPP (in the case where the user has entered some trade- or brand-specific text). The query then performs the search on the relevant terms.

You will note that the query uses a regular expression for the search criteria.[15] This has been used as a concise form of querying any term beginning with (^) or containing a word beginning with ([^a-zA-Z]+), followed by the text “amox”. Additionally, a user-defined parameter “@search_term” has been used for brevity.

9.5.2.5 Query for dispensing use case

The goal of the Dispensing use case in the *AMT v3 Overview and Detailed Business Use Cases* [12] is stated as being

> *“to support using the AMT as the source of the medicines terminology in the dispensing of pack-based prescription by an authorised dispenser (e.g. community pharmacist) and to support the generation and exchange of such information in a community-based model involving an authorised prescriber (e.g. a GP)”.*

In particular, “*the dispensing system shows AMT preferred names using TPPs*”.

In the following example, the prescription is for the AMT concept |*Amoxil 250 mg capsule: hard, 20*|, which is a TPP concept with the Preferred Term “Amoxil 250 mg capsule: hard, 20 capsules”. The prescription indicates that a generic alternative is acceptable.

[15] See https://en.wikipedia.org/wiki/Regular_expression for an introduction.

The dispensing system must find the generic form of the prescribed medication (i.e. the MPP belonging to the prescribed TPP), and then find all TPPs which are associated with that MPP (i.e. all the branded equivalents of the MPP).[16] Using the derived model table **amt_mpp_to_tpp**, a simple query can be written to present a list of appropriate TPPs for dispensing. The query goes further to also return all CTPPs that are associated with those TPPs, as some dispensing systems allow the recording of a CTPP concept or its equivalent.

```
SELECT
    tpp1.tppid AS originaltppid,
    tpp1.tppterm AS originaltppterm,
    substitutetpp.tppid AS substitutetppid,
    substitutetpp.tppterm AS substitutetppterm,
    ctpps.sourceid AS substitutectpp,
    toPt(ctpps.sourceid) AS substitutectppterm

FROM amt_mpp_to_tpp AS tpp1

JOIN amt_mpp_to_tpp AS substitutetpp
     ON tpp1.mppid = substitutetpp.mppid
AND tpp1.tppid != substitutetpp.tppid
AND tpp1.tppid = 12809011000036105 -- Amoxil 250 mg capsule: hard, 20

JOIN relationships_snapshot AS ctpps
     ON substitutetpp.tppid = ctpps.destinationid
AND ctpps.sourceId in (SELECT id FROM amt_ctpp)
ORDER BY originaltppterm, substitutetppterm
;
```

In the above query, the first **JOIN** finds the MPP associated with the prescribed TPP, and the whole range of TPPs associated with these. The second join identifies the corresponding CTPP(s) to the dispenser from the relationships table.

9.5.2.6 Query for extracting dose form

An MPP forms the aggregation of one or more MPUUs, with the addition of pack quantities. Each MPUU has a |*manufactured dose form*| associated with it. While the extraction of dose form does not directly address the core use cases for v3, it may help decision support at the time of both prescribing and dispensing.

The query below seeks to extract each of the substances associated with the MPUUs contained within four sample MPPs. For each substance, the query returns its manufactured dose form for that MPUU.

Note that for some products the TPUU *Form* is different (more specific) to the MPUU *Form*. The query below can be amended to take into account TPUU HAS MANUFACTURED DOSE FORM relationships instead.

[16] Note that AMT does not provide bio-equivalence.

```
SELECT
    amt_ingredient_strength.mppid,
    amt_ingredient_strength.mppterm,
    amt_ingredient_strength.bossterm,
    get_PT(hasDoseForm.destinationid)
FROM amt_ingredient_strength

JOIN relationships_snapshot AS hasDoseForm
     ON hasDoseForm.sourceid = amt_ingredient_strength.mpuuid
     AND hasDoseForm.typeid = 30523011000036108 -- has manufactured dose form
(relationship type)
     AND hasDoseForm.active = 1

WHERE amt_ingredient_strength.mppid in (
    26624011000036107, -- 'amoxycillin 100 mg/mL oral...'
    51572011000036101, -- 'goserelin 3.6 mg implant [...'
    26781011000036107, -- 'peginterferon alfa-2a 135 ...'
    28051011000036109  -- 'peginterferon alfa-2b 150 ...'
)
;
```

Table 11: Results from a sample query that extracts MPP concepts and their associated dose forms

MPP ID	MPP PT	Substance	Dose Form
26624011000036107	amoxycillin 100 mg/mL oral liquid: powder for, 20 mL	amoxycillin	oral liquid: powder for
26781011000036107	goserelin 3.6 mg implant [1] (&) bicalutamide 50 mg tablet [28], 1 pack	goserelin	implant
26781011000036107	goserelin 3.6 mg implant [1] (&) bicalutamide 50 mg tablet [28], 1 pack	bicalutamide	tablet
28051011000036109	peginterferon alfa-2a 135 microgram/0.5 mL injection [4 x 0.5 mL syringes] (&) ribavirin 200 mg tablet [112], 1 pack	peginterferon alfa-2a	injection
28051011000036109	peginterferon alfa-2a 135 microgram/0.5 mL injection [4 x 0.5 mL syringes] (&) ribavirin 200 mg tablet [112], 1 pack	ribavirin	tablet
51572011000036101	peginterferon alfa-2b 150 microgram injection [4 x 150 microgram cartridges] (&) ribavirin 200 mg capsule [196 capsules] (&) inert substance diluent [4 x 0.5 mL cartridges], 1 pack	peginterferon alfa-2b	injection
51572011000036101	peginterferon alfa-2b 150 microgram injection [4 x 150 microgram cartridges] (&) ribavirin 200 mg capsule [196 capsules] (&) inert substance diluent [4 x 0.5 mL cartridges], 1 pack	ribavirin	capsule

10 Subsumption queries

10.1 Subsumption overview

The concepts in SNOMED CT-AU and the AMT are organised in hierarchies of increasing specificity, where concepts at the top are expansive classes, and those at the bottom represent more specific concepts. Specificity increases through the tree, with concepts joined by IS A relationships[17] being more specific.

A subsumption relationship is the most fundamental form of an association between two concepts. It identifies that one concept IS A kind of another concept. All the concepts in the terminologies form a subsumption hierarchy, with a parent concept associated to each child concept through an IS A relationship.

It is straightforward to identify immediate subtype/supertype relationships using the relationships table, as these exist as IS A relationships in the file. However, part of the authoring process of SNOMED CT-AU and the AMT involves classifying the terminology, which includes determining the minimum set of IS A relationships required. Consequently, this set of relationships can change between releases due to the introduction of new content, so it is often more useful to look at all subtype concepts (rather than just direct descendants).

Section 7.7 "Testing and traversing subtype relationships" of the *SNOMED TIG* [1] describes many aspects of using these relationships to determine if one concept is a subtype of another. Of specific interest is the use of a transitive closure table, as described in Section 7.7.5.2 "Transitive Closure Implementation", which also includes code for producing a transitive closure table.

A transitive closure table provides a fast and easy way for determining if two concepts have a subtype/supertype relationship.

[17] IS A relationships are concepts, like everything else in SNOMED CT, in this instance 116680003 |*Is a*|.

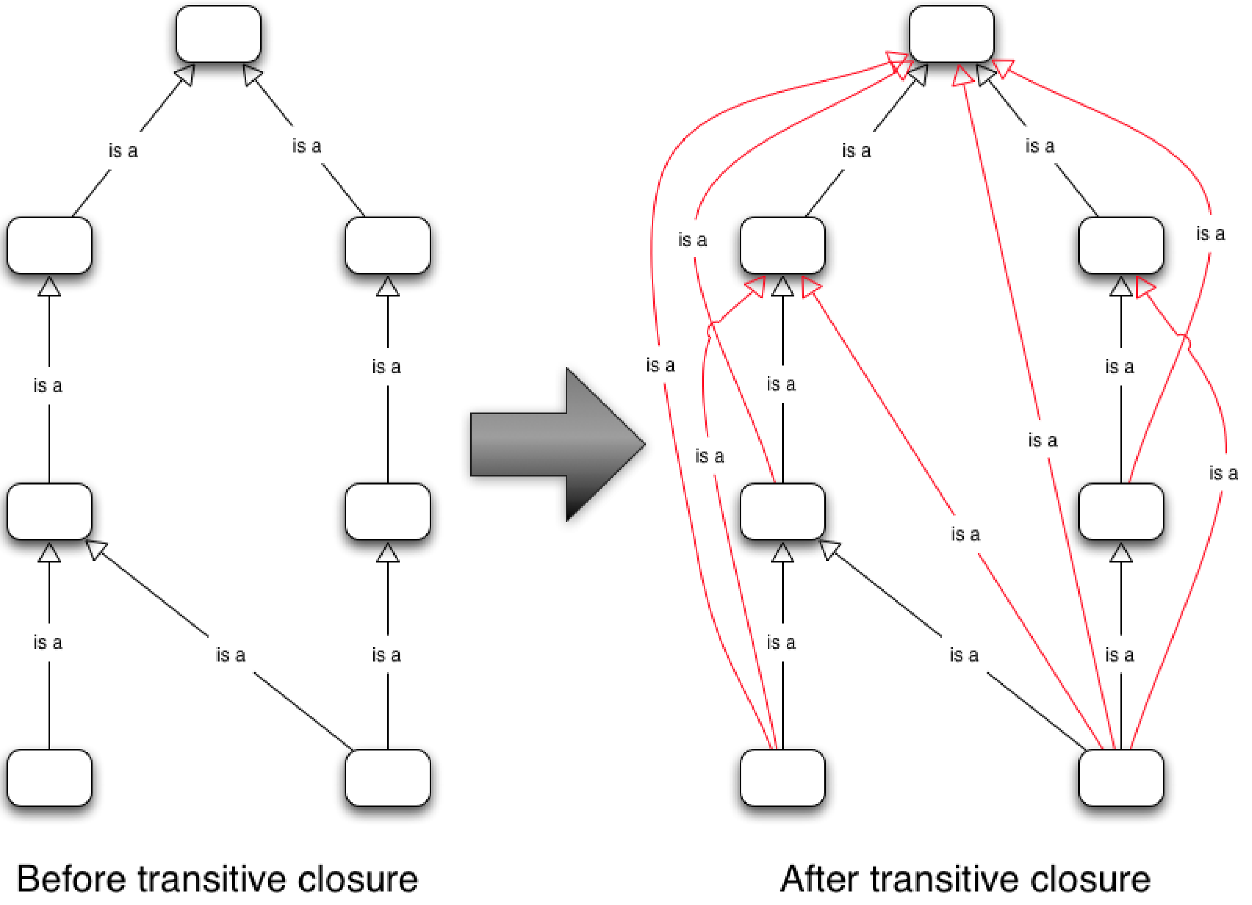


Figure 19:Example of a transitive closure

Upon creation of a transitive closure table, the *distant* subtype/supertype relationships between concepts can be queried in much the same way as child/parent relationships are in the distributed relationship table. A script for generating a transitive closure table and associated indexes is provided in Appendix A.

10.2 SQL examples

10.2.1 Retrieving ancestors using transitive closure

Using the concept 442183006 |*Psychogenic dyskinesia*| as an example, the immediate ancestors (parents) of this concept can be identified directly from the relationship file using:

```
SELECT sourceId,destinationId
FROM relationships_snapshot
WHERE typeId = 116680003
AND sourceId = 442183006;
```

This query shows the two direct ancestral IS A relationships of *Psychogenic dyskinesia*, with the following results.

Table 12: Distributed IS A relationships for Psychogenic dyskinesia

sourceId	destinationId
Psychogenic dyskinesia	*Dyskinesia*
Psychogenic dyskinesia	*Psychophysiologic disorder*

Using the transitive closure table, a similar query such as the following will return a larger set of 17 IS A relationships to all the ancestors of the concept.

```
SELECT sourceId,destinationId
FROM  transitive_closure
WHERE sourceId = 442183006;
```

The query results are tabulated below. Note that the query does not specify a relationship typeId, as all relationships are of the IS A type in the transitive closure table.

Table 13: Transitive closure IS A relationships for Psychogenic dyskinesia

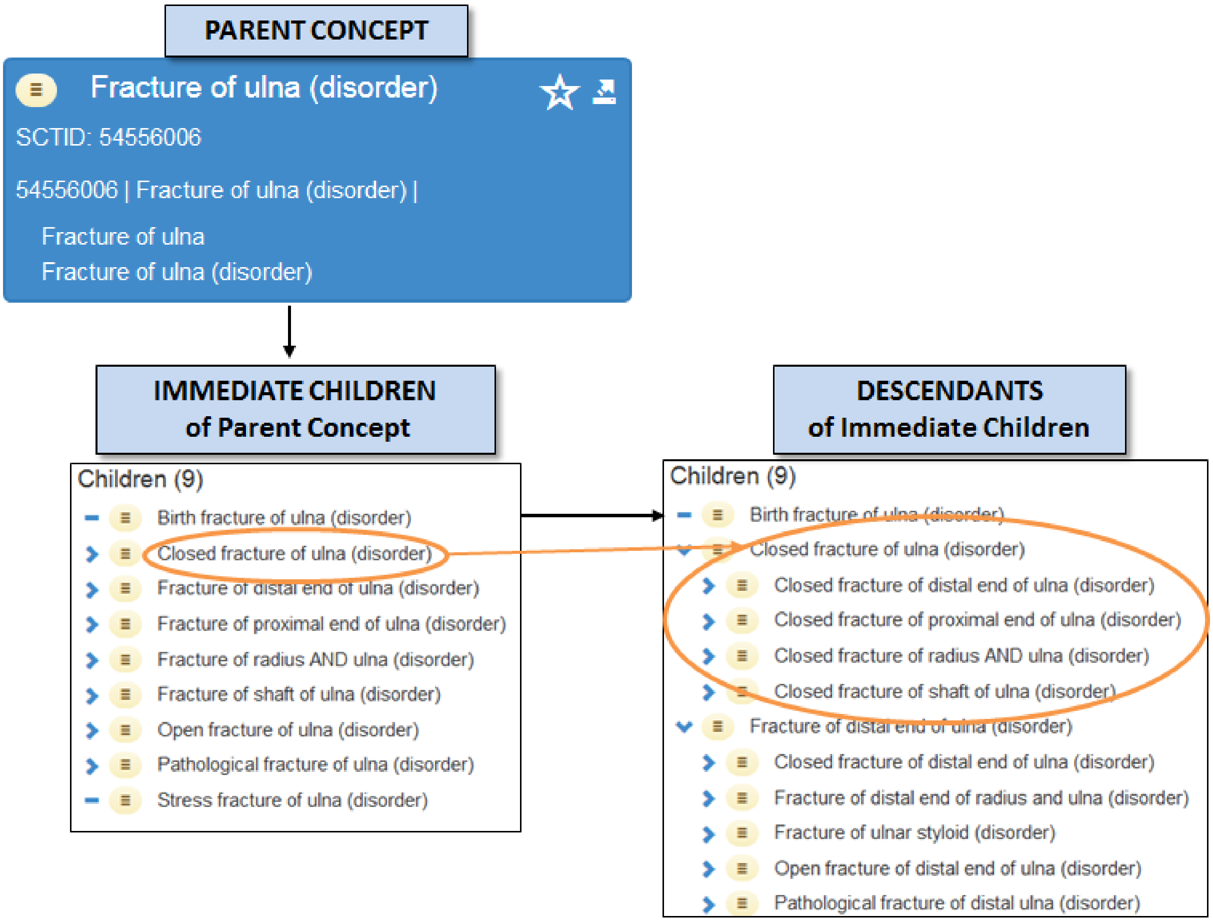
sourceId	destinationId
Psychogenic dyskinesia	*Dyskinesia*
Psychogenic dyskinesia	*Psychosomatic factor in physical condition*
Psychogenic dyskinesia	*Mental state finding*
Psychogenic dyskinesia	*Psychophysiologic disorder*
Psychogenic dyskinesia	*Motor dysfunction*
Psychogenic dyskinesia	*Neurological finding*
Psychogenic dyskinesia	*Perception AND/OR perception disturbance*
Psychogenic dyskinesia	*Motor nervous system finding*
Psychogenic dyskinesia	*Psychological finding*
Psychogenic dyskinesia	*Functional finding*
Psychogenic dyskinesia	*Finding by site*
Psychogenic dyskinesia	*SNOMED CT Concept*
Psychogenic dyskinesia	*Psychological finding of perception*
Psychogenic dyskinesia	*Clinical history and observation findings*
Psychogenic dyskinesia	*Finding of movement*
Psychogenic dyskinesia	*Mental state, behaviour and/or psychosocial function finding*
Psychogenic dyskinesia	*Clinical finding*

10.2.2 Retrieving descendants using transitive closure

To identify the set of concepts that are subsumed by a given concept, a similar approach to that described in Section 10.2.1 can be used. Consider the concept 54556006 |*Fracture of ulna*|. In order to identify all the descendants of this concept, query the transitive closure for all (sourceId) concepts where the destinationId is 54556006.

```
SELECT sourceId,destinationId
FROM transitive_closure
WHERE destinationId = 54556006;
```

This query returns 51 concepts that are subtypes of this concept including the nine immediate children.[18] The diagram below illustrates the query results starting from the parent concept to the immediate child concepts and ending with an expansion of a couple of the direct descendants.


PARENT CONCEPT
IMMEDIATE CHILDREN of Parent Concept
DESCENDANTS of Immediate Children

Fracture of ulna (disorder)

SCTID: 54556006

54556006 | Fracture of ulna (disorder) |

Fracture of ulna
Fracture of ulna (disorder)

Children (9)

- Birth fracture of ulna (disorder)
- Closed fracture of ulna (disorder)
- Fracture of distal end of ulna (disorder)
- Fracture of proximal end of ulna (disorder)
- Fracture of radius AND ulna (disorder)
- Fracture of shaft of ulna (disorder)
- Open fracture of ulna (disorder)
- Pathological fracture of ulna (disorder)
- Stress fracture of ulna (disorder)

Children (9)

- Birth fracture of ulna (disorder)
- Closed fracture of ulna (disorder)
  - Closed fracture of distal end of ulna (disorder)
  - Closed fracture of proximal end of ulna (disorder)
  - Closed fracture of radius AND ulna (disorder)
  - Closed fracture of shaft of ulna (disorder)
- Fracture of distal end of ulna (disorder)
  - Closed fracture of distal end of ulna (disorder)
  - Fracture of distal end of radius and ulna (disorder)
  - Fracture of ulnar styloid (disorder)
  - Open fracture of distal end of ulna (disorder)
  - Pathological fracture of distal ulna (disorder)

Figure 20: Descendants of "Fracture of ulna"

[18] As of SNOMED CT-AU February 2016. Specific numbers may vary for other releases.

10.2.3 Retrieving complex sets of descendants with exclusions

More complex sets of concepts can be identified by combining multiple subsumption queries and exclusion criteria. One use case is where there is a smaller set of concepts that need to be excluded from a greater set of concepts. As an example, there are 288 concepts of type 399907009 |*Animal bite wound*|. Among these are 172 concepts of type 409985002 |*Arthropod bite wound*|.

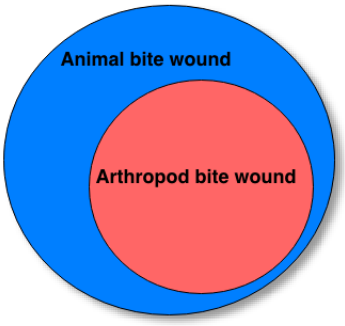


Figure 21: Relationship between Animal bite wound and Arthropod bite wound concepts

The following query can be used to identify the set of *Animal bite wound* concepts, excluding the *Arthropod bite wounds*.

```
SELECT sourceId
FROM transitive_closure
WHERE destinationId = 399907009 -- Animal bite wound
-- exclude the concepts that are 409985002 |Arthropod bite wound| descendants
AND sourceId NOT IN (SELECT sourceId
                     FROM transitive_closure
                     WHERE destinationId = 409985002
                     -- Arthropod bite wound
                     );
```

This query provides the set of 116 concepts that are not *Arthropod bite wounds*.

The same approach can be used to determine the relative complement between two sets of concepts.

10.2.4 Retrieving the intersection of two subhierarchies

Concepts in SNOMED CT-AU are organised in a polyhierarchy[19], and may be descendants of more than one, disjoint concept. As an example there are over:

- 400 descendants of 271737000 |*Anaemia*|; and
- Six thousand 66091009 |*Congenital disease*| concepts.

[19] Concepts in SNOMED CT-AU may have more than one parent concept and, as a consequence, many sub-hierarchies contain overlapping content.

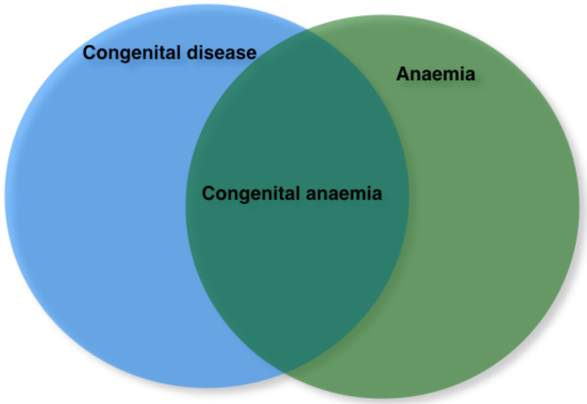


Figure 22: Intersection of Congenital diseases and Anaemias

An **INNER JOIN** query is very well suited to identifying the intersection, as illustrated by the following query:

```
SELECT anaemia.sourceId
FROM transitive_closure AS anaemia
INNER JOIN transitive_closure AS congenital
ON anaemia.sourceId = congenital.sourceId
WHERE anaemia.destinationId = 271737000 -- | anaemia |
AND congenital.destinationId = 66091009 -- | congenital disease |
;
```

Another approach is to apply a similar query to that described in Section 10.2.3:

```
SELECT sourceId
FROM transitive_closure AS anaemia
WHERE anaemia.destinationId = 271737000 -- | anaemia |
AND sourceId IN (
             SELECT sourceId FROM transitive_closure AS congenital
             WHERE congenital.destinationId = 66091009 -- | congenital
disease |
);
```

Each of the above two queries will return the same set of 44 *Congenital anaemia* concepts.

11 Maintenance of Terminology updates

Each SNOMED CT-AU and AMT release includes changes to the terminology content. Some of the reasons for which content changes are required are to:

- correct errors and fill gaps;
- stay abreast of changing clinical knowledge;
- map to or align with other code systems and classifications.

Regardless of implementation type or technologies, terminology content will have to be regularly updated within an implementation. Under the terms of the Australian National Terminology Release Licence[20], an implementation is required to update to a new version of SNOMED CT-AU and the AMT at a minimum of every 180 days. However, business needs may require more frequent updates, especially for systems supporting medicinal concepts.

Requests for further terminology and support queries should be directed to help@nehta.gov.au.

We will continue to develop and enhance new and existing reference sets to satisfy identified terminology requirements. Stakeholder feedback on reference set content is a vital factor that will facilitate further development.

NEHTA can assist with some of the steps along the implementation path by:

- Providing tools for viewing SNOMED CT-AU and its content, such as reference sets.
- Offering expert skills and education to SNOMED CT-AU users.
- Engaging with the wider SNOMED CT and implementation community.
- Publishing documented experiences to provide advice on models for implementation to the community.
- Hosting reference sets owned and developed by the community.

Additions or modifications to existing terminology content may be requested by using Request Submission forms, accessible at https://www.nehta.gov.au/get-started-with-ehealth/what-is-ehealth/clinical-terminology/request-submission-product-content-changes

11.1 Types of changes

Changes can be to concepts, descriptions, relationships and reference sets in the form of additions, inactivations and reactivations and changes to concept status.

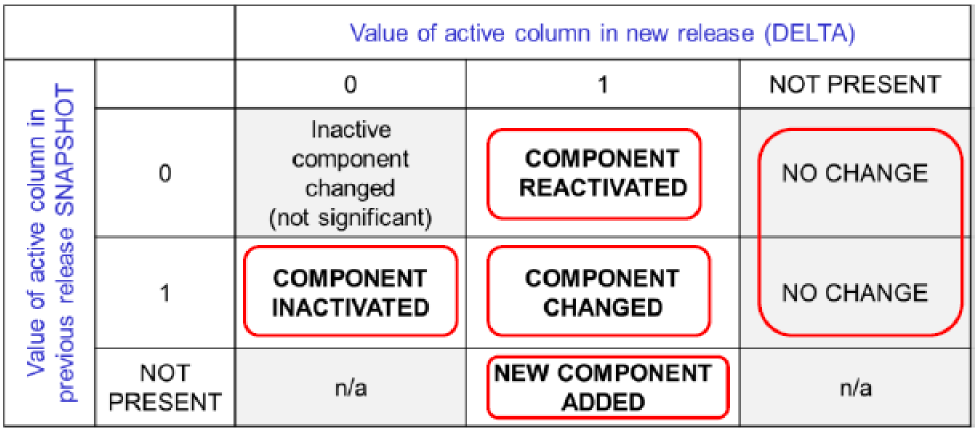
The amendments of reference sets involve changes to the individual members of the reference set. These changes would include adding and removing members, rather than adding or removing entire reference sets. The effects of these changes depend on the type of reference set and the way in which it is used.

[20] https://www.nehta.gov.au/get-started-with-ehealth/what-is-ehealth/clinical-terminology/registering-for-a-license

11.2 Identifying changes

In order to identify content changes that have occurred for a particular release, a Delta release type can be compared against the Snapshot view from the previous release.

Changes to terminology content can be identified through the various types of the release files, namely Full, Snapshot and Delta.



		Value of active column in new release (DELTA)		
		0	1	NOT PRESENT
Value of active column in previous release SNAPSHOT	0	Inactive component changed (not significant)	COMPONENT REACTIVATED	NO CHANGE
	1	COMPONENT INACTIVATED	COMPONENT CHANGED	NO CHANGE
	NOT PRESENT	n/a	NEW COMPONENT ADDED	n/a

Figure 23: Understanding content changes through the Snapshot and Delta files

11.3 Potential impact of terminology changes

The impact of changes to terminology content may affect:

- Data entry protocols and terminology bindings in a user interface.
- Communication specifications with external systems.
- Links to knowledge resources.
- Results of queries used for reporting and analytics.

The following table summarises how various types of changes can be managed in a clinical information system.

Table 14: Managing changes in clinical information systems

	Data entry	**Communications with external systems**	**Knowledge resources**	**Reporting and analytics**
Added concepts	Relevant value sets, pick lists, data entry templates or interface terminology maps need to be updated.	Message specifications and any maps to or from code systems used to enable communication may need to be updated.	Consider adding new concepts to trigger conditions for decision support and knowledge links.	Consider adding new concepts to relevant query filters and maps to classifications used for reporting.
Inactivated concept	Remove or replace concepts from value sets, pick lists, data entry templates or interface terminology maps.	Remove or replace inactivated concepts from message specifications and any maps to or from code systems used to enable communication.	Remove or replace inactivated concepts to trigger conditions for decision support and knowledge links.	Remove or replace inactivated concepts in query filters, subsets and maps to classification used for reporting.

	Data entry	Communications with external systems	Knowledge resources	Reporting and analytics
Added/inactivated Relationship	As change affects subsumption and concept definitions, check whether changes to intensional value set definitions are appropriate to their intended use.	As change affects subsumption and concept definitions, check whether changes to intensional value set definitions used in message specifications are appropriate.	As change affects subsumption and concept definitions, check whether changes to intensionally defined decision support triggers or links are appropriate.	Consider amending queries if the overall effect of the updates on the results is not appropriate.
Inactivated Description	Check description use in data entry lists or screen inputs. Replace them with appropriate active descriptions.			

Appendix A Transitive closure script

As described in Section 10, transitive closure presents an expanded view of all possible IS A relationships contained within the terminology. For example, if concept **α** IS A concept **β**, and concept **β** IS A concept **γ**, then it can be inferred that concept **α** IS A concept **γ**, even if that relationship is not explicitly stated. The transitive closure table includes a row for each of these inferred relationships, as well as including all explicitly stated IS A relationships.

A procedure for creating that table is provided below. The syntax for this procedure is specific to the MySQL database platform and will likely require modification if used with other platforms. The transitive closure can be used for subsumption queries, for example, to find all descendants of 40733004 |*Infectious disease (disorder)*|.

The following is a script to create a stored procedure that will generate a transitive closure table. The script has been adapted from the one provided in Section 7.7.5.2 "Transitive Closure Implementation" in the *SNOMED TIG* [1].

A.1 SQL Script to create the Transitive Closure table

```
/* -------------------------------------------------------------------
Demonstration Transitive Closure creation script
Note an alternative (though more verbose) script is available in the
SNOMED CT Technical Implementation Guide:

www.snomed.org/tig?t=tsg2_test_optimizeConcept_transitiveClosure_impl
_generate
--------------------------------------------------------------------*/

DELIMITER //

DROP PROCEDURE IF EXISTS createTransitiveClosure //
CREATE PROCEDURE createTransitiveClosure ()

BEGIN
-- Create the Transitive Closure table schema
    DROP TABLE IF EXISTS Transitive_Closure;
    CREATE TABLE Transitive_Closure (
            sourceid BIGINT NOT NULL,
            destinationid BIGINT NOT NULL,
            PRIMARY KEY (sourceid, destinationid)
            ) ENGINE = MyISAM;

-- Insert the immediate set of IS A relationships from the distributed
relationships table
    INSERT INTO Transitive_Closure (sourceid,destinationid)
           SELECT DISTINCT sourceid,destinationid
           FROM relationships_snapshot
           WHERE typeid = 116680003 -- "IS A" relationship type
           AND active = 1;

-- Recursively loop through the transitive closure adding additional
relationships until there are no more left to insert
    REPEAT

    INSERT INTO Transitive_Closure (sourceid,destinationid)
                SELECT DISTINCT b.sourceid,a.destinationid
                FROM Transitive_Closure a
                JOIN Transitive_Closure b
                ON a.sourceid = b.destinationid
                LEFT JOIN Transitive_Closure c
                ON c.sourceid = b.sourceid
                AND c.destinationid = a.destinationid
                WHERE c.sourceid IS NULL;

    SET @x = row_count();
-- Non essential output logger.
    SELECT CONCAT ('Inserted ',@x);

    UNTIL @x = 0
END REPEAT;

CREATE INDEX TC_sourceid_idx ON Transitive_Closure (sourceid);
CREATE INDEX TC_destinationid_idx ON Transitive_Closure (destinationid);

END //
```

```
/* -------------------------------------------------------------------
Demonstration Transitive Closure creation script
```

```
Note an alternative (though more verbose) script is available in the
SNOMED CT Technical Implementation Guide:

www.snomed.org/tig?t=tsg2_test_optimizeConcept_transitiveClosure_impl_
generate
----------------------------------------------------------------------*/

DELIMITER //

DROP PROCEDURE IF EXISTS createTransitiveClosure //
CREATE PROCEDURE createTransitiveClosure ()

BEGIN
-- Create the Transitive Closure table schema
    DROP TABLE IF EXISTS Transitive_Closure;
    CREATE TABLE Transitive_Closure (
            sourceid BIGINT NOT NULL,
            destinationid BIGINT NOT NULL,
            PRIMARY KEY (sourceid, destinationid)
            ) ENGINE = MyISAM;

-- Insert the immediate set of IS A relationships from the distributed
relationships table
    INSERT INTO Transitive_Closure (sourceid,destinationid)
           SELECT DISTINCT sourceid,destinationid
           FROM relationships_snapshot
           WHERE typeid = 116680003 -- "IS A" relationship type
           AND active = 1;

-- Recursively loop through the transitive closure adding additional
relationships until there are no more left to insert
    REPEAT

    INSERT INTO Transitive_Closure (sourceid,destinationid)
            SELECT DISTINCT b.sourceid,a.destinationid
            FROM Transitive_Closure a
            JOIN Transitive_Closure b
            ON a.sourceid = b.destinationid
            LEFT JOIN Transitive_Closure c
            ON c.sourceid = b.sourceid
            AND c.destinationid = a.destinationid
            WHERE c.sourceid IS NULL;

    SET @x = row_count();
-- Non essential output logger.
    SELECT CONCAT ('Inserted ',@x);

    UNTIL @x = 0
END REPEAT;

CREATE INDEX TC_sourceid_idx ON Transitive_Closure (sourceid);
CREATE INDEX TC_destinationid_idx ON Transitive_Closure
(destinationid);

END //
```

Appendix B Relationship between SNOMED CT-AU and the AMT

The AMT is a SNOMED CT terminology. It shares the same technical format and top level hierarchy as SNOMED CT-AU, and hence the same "semantic space".

B.1 Modules and dependencies

One of the features introduced with the release format is modules, which enable relatively complex relationships and activities such as:

- ownership of components to transfer between organisations, while maintaining the same identifiers;
- assembly of custom editions; and
- identification of module dependencies.

A SNOMED CT module is defined in the *SNOMED TIG* [1] as:

> *A group of SNOMED CT components and/or reference set members that are at a given point in time managed, maintained and distributed as a unit.*[21]

As an example, the content for the International Release is distributed across two modules:

- SNOMED CT core – the clinical content of SNOMED CT.
- SNOMED CT model component – the metadata required to define the release format.

Similarly the two terminologies produced by NEHTA, namely SNOMED CT-AU and the AMT, are composed of modules as illustrated below.

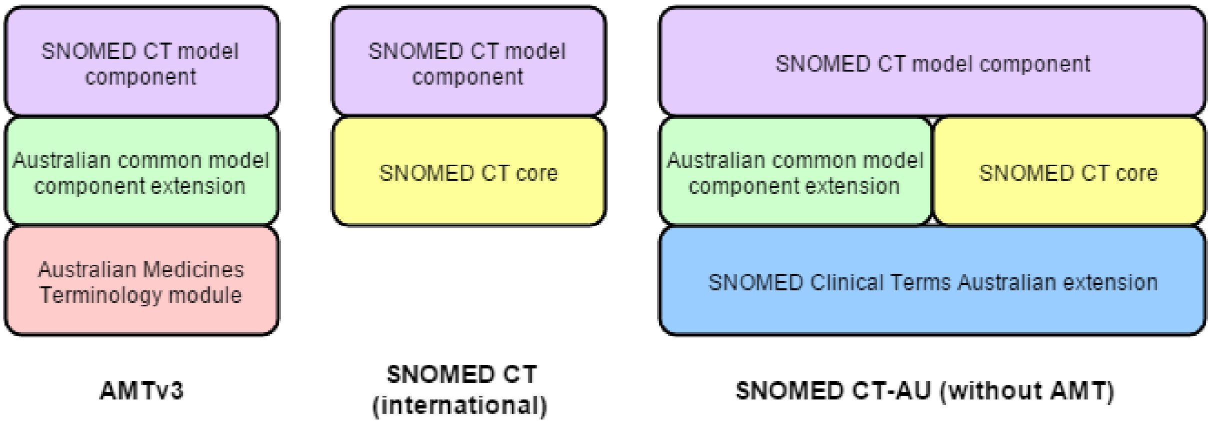

SNOMED CT model component
Australian common model component extension
Australian Medicines Terminology module
AMTv3
SNOMED CT model component
SNOMED CT core
SNOMED CT (international)
SNOMED CT model component
Australian common model component extension
SNOMED CT core
SNOMED Clinical Terms Australian extension
SNOMED CT-AU (without AMT)

Figure 24: Release modules in terminology releases

The *Module dependency reference set*[22] specifies the relationship between modules, and in the case of the International Release, shows that the clinical content module (*core*) is dependent on the metadata module (*model component*). It is important to note that each release (version) of a module is dependent on a specific set of versions of other modules.

[21] *IHTSDO Glossary* [16].

[22] Section 7.4.2.4 "Module Dependency Reference Set" of the *SNOMED TIG.*

Consequently the module dependency is updated each release. The November 2014 SNOMED CT-AU release has a dependency chain as depicted in the following figure.

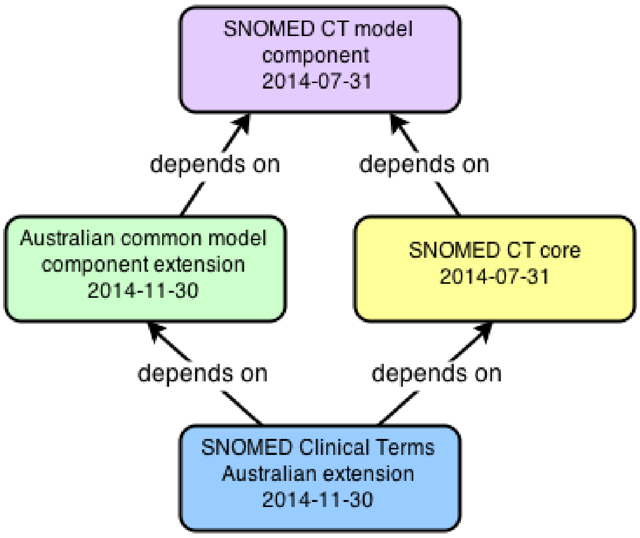

SNOMED CT model component 2014-07-31
depends on
depends on
Australian common model component extension 2014-11-30
SNOMED CT core 2014-07-31
depends on
depends on
SNOMED Clinical Terms Australian extension 2014-11-30

Figure 25: SNOMED CT-AU module dependency as of November 2014

Whenever a module is changed, either by adding or modifying components, the result is a new version module. The module dependencies are version specific, and thus the dependencies are updated every time a module is modified.

B.2 Combining SNOMED CT-AU and the AMT

Figure 26 below shows the modular nature of NCTIS terminology releases. It can also be seen that there are some modules common to both SNOMED CT-AU and the AMT. As of November 2015, the NCTIS is including the AMT as a part of SNOMED CT-AU. The benefit of this is that vendors who use both products only need to download and manage a single terminology.

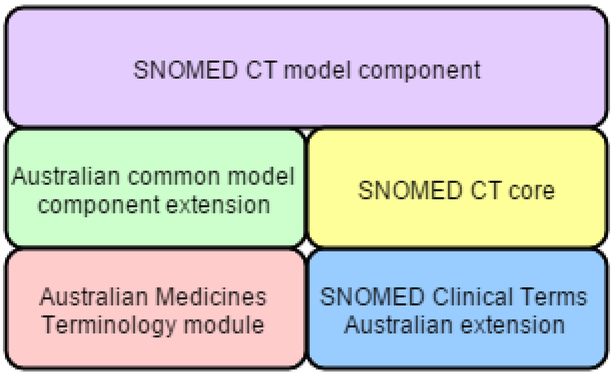

SNOMED CT model component
Australian common model component extension
SNOMED CT core
Australian Medicines Terminology module
SNOMED Clinical Terms Australian extension

Figure 26: Module composition for SNOMED CT-AU combined with the AMT

Content from the AMT module can be identified in all the tables as having a value in the **moduleId** field of 900062011000036108. There is no difference in the AMT content between the combined SNOMED CT-AU and stand-alone distributions. The technical design also means that existing users of either terminology should not need to make any changes to their systems to use the combined release. As the AMT becomes more tightly integrated with SNOMED CT-AU in future releases, some changes may become necessary. The NCTIS will highlight any such changes in its communications.

Appendix C Metadata cheat sheet

C.1 Description types

Name of description type	Description type identifier
Fully Specified Name	900000000000003001
Acceptable Synonym	900000000000013009

C.2 ADRS preferences

Name of description type	Acceptability
Preferred	900000000000548007
Acceptable	900000000000511003

C.3 Relationship types

Name of relationship	Relationship identifier
ACCESS	260507000
ASSOCIATED FINDING	246090004
ASSOCIATED MORPHOLOGY	116676008
ASSOCIATED PROCEDURE	363589002
ASSOCIATED WITH	47429007
CAUSATIVE AGENT	246075003
CLINICAL COURSE	263502005
COMPONENT	246093002
DIRECT DEVICE	363699004
DIRECT MORPHOLOGY	363700003
DIRECT SUBSTANCE	363701004
DUE TO	42752001
FINDING CONTEXT	408729009
FINDING INFORMER	419066007
FINDING METHOD	418775008
FINDING SITE	363698007
FOLLOWING	255234002
FOLLOWING	255260001

Name of relationship	**Relationship identifier**
HAS ACTIVE INGREDIENT	127489000
HAS AUSTRALIAN BoSS	30364011000036101
HAS COMPONENT PACK	700000061000036106
HAS CONTAINER TYPE	30465011000036106
HAS DEFINITIONAL MANIFESTATION	363705008
HAS DENOMINATOR UNITS	700000071000036103
HAS DOSE FORM	411116001
HAS FOCUS	363702006
HAS INTENDED ACTIVE INGREDIENT	700000081000036101
HAS INTENT	363703001
HAS INTERPRETATION	363713009
HAS MANUFACTURED DOSE FORM	30523011000036108
HAS MPUU	30348011000036104
HAS NUMERATOR UNITS	700000091000036104
HAS SPECIMEN	116686009
HAS SUBPACK	30454011000036104
HAS TP	700000101000036108
HAS TPUU	30409011000036107
HAS UNIT OF USE	30548011000036101
INDIRECT DEVICE	363710007
INDIRECT MORPHOLOGY	363709002
INTERPRETS	363714003
IS A	116680003
IS MODIFICATION OF	30394011000036104
LATERALITY	272741003
MEASUREMENT METHOD	370129005
METHOD	260686004
OCCURRENCE	246454002
PART OF	123005000
PATHOLOGICAL PROCESS	370135005
PRIORITY	260870009
PROCEDURE CONTEXT	408730004
PROCEDURE DEVICE	405815000

Name of relationship	Relationship identifier
PROCEDURE MORPHOLOGY	405816004
PROCEDURE SITE	363704007
PROCEDURE SITE – DIRECT	405813007
PROCEDURE SITE – INDIRECT	405814001
PROPERTY	370130000
RECIPIENT CATEGORY	370131001
REVISION STATUS	246513007
ROUTE OF ADMINISTRATION	410675002
SCALE TYPE	370132008
SPECIMEN PROCEDURE	118171006
SPECIMEN SOURCE IDENTITY	118170007
SPECIMEN SOURCE MORPHOLOGY	118168003
SPECIMEN SOURCE TOPOGRAPHY	118169006
SPECIMEN SUBSTANCE	370133003
SUBJECT RELATIONSHIP CONTEXT	408732007
SURGICAL APPROACH	424876005
TEMPORAL CONTEXT	408731000
USING ACCESS DEVICE	425391005
USING DEVICE	424226004
USING ENERGY	424244007
USING SUBSTANCE	424361007

Acronyms

Acronym	Description
ADRS	Australian dialect reference set
AMT	Australian Medicines Terminology
DDL	Data Definition Language (also known as “Data Description Language”)
PBD	Pharmaceutical Benefits Division
IHTSDO	International Health Terminology Standards Development Organisation
NCTIS	National Clinical Terminology and Information Service
RF2	SNOMED CT Release Format 2.0
SNOMED CT-AU	SNOMED CT, Australian release
SQL	Structured Query Language
PBS	Pharmaceutical Benefits Scheme
TGA	Therapeutic Goods Administration

References

1. IHTSDO. *SNOMED CT Technical Implementation Guide*. Copenhagen: IHTSDO; 2015. January 2015 release. Available from: https://confluence.ihtsdotools.org/display/DOC.

2. NEHTA. *SNOMED CT-AU - Australian Implementation Guidance*. Sydney: NEHTA; 2015. v2.0. Available from: http://www.nehta.gov.au/implementation-resources/ehealth-foundations/snomed-ct-au-common.

3. NEHTA. *NCTIS Reference set library*. Sydney: NCTIS; 2016. v2.2. Available from: https://www.nehta.gov.au/implementation-resources/ehealth-foundations/snomed-ct-au-common.

4. NEHTA. *NCTIS Development approach for reference sets*. Sydney: NEHTA; 2016. v2.2. Available from: http://www.nehta.gov.au/implementation-resources/ehealth-foundations/snomed-ct-au-common.

5. NEHTA. *Australian Medicines Terminology v3 Model - Technical Implementation Guide*. 2015. v2.1. Available from: http://www.nehta.gov.au/implementation-resources/ehealth-foundations/australian-medicines-terminology-common.

6. NEHTA. *AMT v3 Overview and Detailed Business Use Cases*. Sydney: NEHTA; 2013. v1.0. Available from: http://www.nehta.gov.au/implementation-resources/ehealth-foundations/australian-medicines-terminology-v3-common.

7. NEHTA. *AMT v3 Model Editorial Rules*. Sydney: NEHTA; 2014. v2.0. Available from: http://www.nehta.gov.au/implementation-resources/ehealth-foundations/australian-medicines-terminology-v3-common.

8. NEHTA. *Clinical Terminology - Guidance for People and Processes*. 2014. v1.0. Available from: http://www.nehta.gov.au/implementation-resources/ehealth-foundations/australian-medicines-terminology-common.

9. NEHTA. *Clinical Terminology - Guidance for Use in Healthcare Software*. 2014. v1.0. Available from: http://www.nehta.gov.au/implementation-resources/ehealth-foundations/australian-medicines-terminology-common.

10. NEHTA. *Clinical Terminology - Use of Medical Nomenclatures in Information Exchange*. 2014. v1.0. Available from: http://www.nehta.gov.au/implementation-resources/ehealth-foundations/australian-medicines-terminology-common.

11. NEHTA. *AMT Mapping Guidelines*. Sydney: NEHTA; 2012. v1.1. Available from: http://www.nehta.gov.au/implementation-resources/ehealth-foundations/australian-medicines-terminology-common.

12. IHTSDO. *SNOMED CT Editorial Guide*. Copenhagen: IHTSDO; 2015. January 2015 release. Available from: https://confluence.ihtsdotools.org/display/DOC.

13. IHTSDO. *SNOMED CT URI Standard*. 2014. v1.0. Available from: https://confluence.ihtsdotools.org/display/DOC.

14. NEHTA. *AMT 2012 Survey Results and Development Roadmap*. Sydney: NEHTA; 2012. v1.0. Available from: http://www.nehta.gov.au/implementation-resources/ehealth-foundations/australian-medicines-terminology-v2-common.

15. Pharmaceutical Benefits Scheme. *Pharmaceutical Benefits Scheme (PBS).* [Internet]. 2012 [cited 2013 Feb 22]. Available from: http://www.pbs.gov.au/info/home.

16. IHTSDO. *IHTSDO Glossary*. Copenhagen: IHTSDO; 2014. July 2014 Release. Available from: https://confluence.ihtsdotools.org/display/DOC.