Australian Government
Australian Digital Health Agency

SNOMED CT-AU
Australian Technical Implementation Guide

14 June 2024 v4.0

Document ID: DH-3243-2020

Australian Digital Health Agency ABN 84 425 496 912, Level 25, 175 Liverpool Street, Sydney, NSW 2000
Telephone 1300 901 001 or email help@digitalhealth.gov.au
www.digitalhealth.gov.au

Acknowledgements
The Australian Digital Health Agency is jointly funded by the Australian Government and all state and territory governments.

Disclaimer
The Australian Digital Health Agency ("the Agency") 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. The Agency 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.

OFFICIAL

Document information

Key information

Owner	General Manager Privacy and Compliance
Date of next review	29 August 2025
Contact for enquiries	Australian Digital Health Agency Help Centre Phone 1300 901 001 Email help@digitalhealth.gov.au

Product or document version history

Product or document version	Date	Release comments
1.0	31 May 2016	First release.
1.1	30 Jun 2016	New section: “Dose-based prescribing ingredient selection” (8.45).
2.1	5 Aug 2016	Agency branding, revised name and version number, updated 8.4.5, added 8.4.6, 8.4.7.
2.2	31 Aug 2017	Updated 1.3.3, 4.1, 6.1.1; added 8.5, 8.6, 8.7, 8.8, Appendix D, Appendix E
2.3	30 Apr 2018	New section: “Schedule 8 medicines” (9.5.3);
2.4	30 Jun 2019	Updated 1.5, 1.6, 4.1, 5.1, 6.4, 8.4.2, 9.2, Appendix C; added 8.4.8, 8.4.9
2.5	3 Jul 2020	Updated references from STU3 to R4
4.0		Update to reflect AMTv4 model and minor RF2 changes

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 including 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

SNOMED International provide substantial documentation available through their online Document Directory. Of particular note are the:

- *SNOMED CT* Release File Specification (SNOMED International, 2024)
- *SNOMED CT Editorial Guide* (SNOMED International, 2024), which provides details of the core SNOMED CT concept model and editorial rules.
- SNOMED CT Starter Guide (SNOMED International, 2023)

Australian developers should use this document and the *SNOMED TIG* together. Cross-references to specific sections within the *SNOMED TIG* and other SNOMED International documents are provided at various points in this document.

1.3.2 SNOMED CT-AU

This document does not focus on the development of release artefacts or products. For example, the *SNOMED CT-AU Development Approach for Reference Sets* [3] describes 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 document does not cover a business overview of the AMT or drivers for implementing the AMT in general. These topics are covered by the *AMT Concept Model and Business Use Cases* (Australian Digital Health Agency, 2017).

1.4 Clinical terminology overview

A clinical terminology contributes to the improvement of health care 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 can also support or enable:

- Accurate recording of clinical information at the required level of granularity.
- Semantic interoperability between disparate clinical information systems.
- Reusability of clinical information ("record once, use many times" – secondary usage or analysis)
- Consistent representation of clinical terms.
- Machine processing of clinical information.

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 terminology product. 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, relationship types, extension mechanisms 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 SNOMED CT-AU sample scripts available in the Australian Digital Health Agency GitHub repository[1].

The accompanying query examples herein use a Snapshot release unless otherwise indicated.[2]

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.

[1] Available from https://github.com/AuDigitalHealth/sctau-sample-scripts.

[2] See Section 4.2 for details about release types.

1.6 National Terminology Server and Ontoserver

The content of this document is intended for implementations that consume the terminology releases directly within their systems, often with an internal or bespoke terminology solution. This approach relies on a sound knowledge of the terminology components and release file specifications.

The National Clinical Terminology Service (NCTS) now also provides an alternative solution in the form of Ontoserver®.[3] Ontoserver is a terminology server developed by the CSIRO that supports multiple versions of SNOMED CT (and extensions), LOINC and other standard terminologies. It also supports the HL7™ FHIR® standard. The advantage of this solution is that many of the complexities of maintaining a terminology database are handled by the server.

The National Terminology Server (NTS) is the Agency's deployment of Ontoserver, hosting national terminology content. For availability and performance reasons it is not suited to runtime system integration, but it can be used to extract data for packaging with software. The FHIR® resources published by the NCTS are available for browsing[4] as well as Postman collections[5] that demonstrate FHIR® operations.

The National Syndication Server (NSS) is the primary source of national terminology content. It exposes a programmatic Syndication Interface which enables content downloads to be automated. Both RF2 release bundles and prebuilt Ontoserver binary indexes are available in the feed. More information is available from the NCTS website.

Ontoserver is available for free for use by registered licence holders in Australia. Exemplars using this technology are published in open-source repositories[6]. Further details are available at the NCTS website > Tools, in the NCTS Guide for Implementers (Australian Digital Health Agency, 2017) document, or contact help@digitalhealth.gov.au.

[3] Ontoserver is a registered trademark of the Commonwealth Scientific and Industrial Research Organisation (CSIRO).
[4] Available from https://healthterminologies.gov.au/fhir/CodeSystem, https://healthterminologies.gov.au/fhir/ValueSet, and https://healthterminologies.gov.au/fhir/ConceptMap.
[5] Available from https://www.healthterminologies.gov.au/tools/national-terminology-server/index.html and https://ontoserver.csiro.au/docs/5.3/
[6] Available from https://audigitalhealth.github.io/ecl-examples/ and https://aehrc.github.io/fhir-ts-exemplars/.

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 created at the request of Australian users to support implementations. Extension content can include additional terms for existing concepts, new concepts and reference sets. Content is often created collaboratively in small projects with external stakeholders including jurisdictions, vendors and clinicians.

The SNOMED CT-AU release bundle is published monthly and contains the relevant SNOMED CT International components as well as the content developed by the Agency for use in Australian healthcare 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 the Agency 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, and 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 consistent naming conventions to identify branded and generically equivalent medicines; and
- describing medicines in a way that that may be programmatically accessed and managed within clinical information systems.

The Agency 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 every month; it is based on updates from the Pharmaceutical Benefits Scheme (PBS) and Therapeutic Goods Administration (TGA).

The AMT is published as part of SNOMED CT-AU.

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 the products in the categories listed above. The addition of new products is prioritised based on feedback from end users.[7]

The structure of the AMT is based upon “Products” and “Packs”. Product concepts describe either continuant (liquid) or discrete unit (tablets) of medicinal products or devices. Pack concepts describe specific quantities of these products. Within each of these are concepts of varying levels of specificity. The “seven notable reference sets” are used to identify concepts across the terminology with various levels of specificity so that the required content can be identified and implemented for different contexts.

- Containered Trade Product Pack (CTPP)– These concepts represent the most specific type of concept- branded packs of branded products, in a specific container/packaging. These represent a real marketable product down to the specific container (e.g. bottle or blister pack).
- Trade Product Pack (TPP) – Much like CTPPs but without a specific container type. These are “branded packs of branded products”
- Trade Product Unit of Use (TPUU) – These are the branded products within a CTPP (or TPP). They could be countable discrete units like tablets or ampoules, or unbound continuants such as liquid or cream. Details at this level include specific strengths of active ingredients, manufactured form, and if required container types and size.
- Trade Products (TPs) – Are essentially “Brands”. These concepts provide specific product names (or labels), but no specific modelling around form, ingredients or strengths.
- Medicinal Product Pack (MPP) – Like TPPs but are brand agnostic.
- Medicinal Product Units of Use (MPUUs) – Like TPUUs but brand agnostic.
- Medicinal Products (MPs) – These are the simplest level detail, providing just an abstraction to just a combination of ingredients. Devoid of form or strength details.

Within AMTv4 these 7 levels of abstraction are identified through the “7 notable reference sets” (named as above). The concepts within each of these reference sets are modelled using defining properties and organised within the greater SNOMED CT-AU terminology. The concepts have both hierarchical and other non-hierarchical relationships to each other and other concepts in the terminology.

The names and acronyms for these reference sets carry over from AMTv3. And while there is no longer a “TPUU” concept class in AMTv4, “TPUUs” is use to describe members of the TPUU reference set – as they are describing the same level of granularity.

[7] See the NCTS request submission page.

The concepts are organised within SNOMED CT-AU as a hierarchy such that concepts become more specific progressively down the tree. 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.

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 Agency governance body or bodies.

A Medicinal Product will only define inactive (inert) ingredients where these are part of sequential multi-component 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 uses concept models for different hierarchies of content to 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 of 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 can 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.

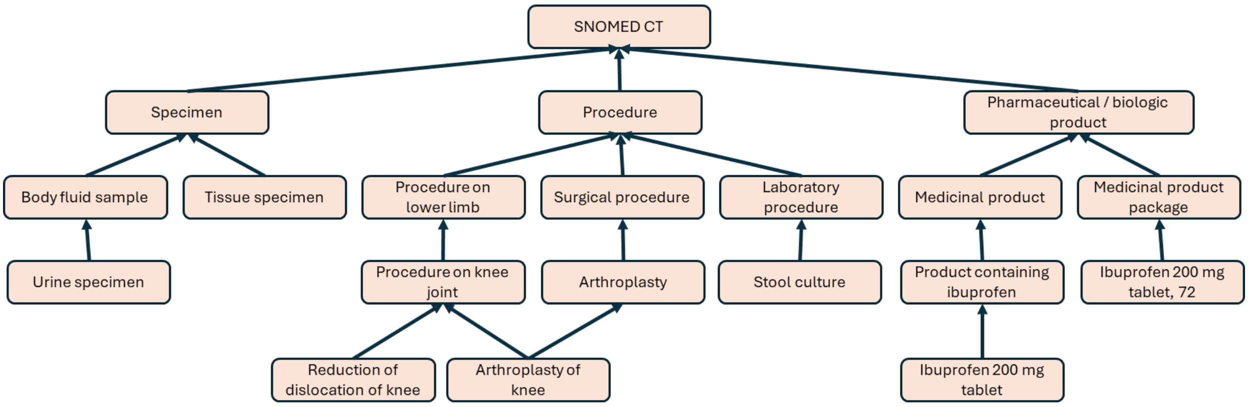


Figure 1: Example conceptual view of the SNOMED CT AU hierarchy

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 (FSN), 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.

Further information about these components is available in (SNOMED CT Release File Specifications) – High Level Logical Model[8].

3.2 AMT concept model

Figure 2 below provides an illustrative summary of the AMT v4 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 AMT v4 concept model are described in the *AMT v4 Model Specification* (Australian Digital Health Agency, 2024)
- Full details of the target use cases that the AMT supports are described in the *AMT Concept Model and Business Use Cases* (Australian Digital Health Agency, 2017).
- The naming conventions of AMT descriptions and full definitions of the AMT classes are described in the *AMT v4 Editorial and Naming Rules* (Australian Digital Health Agency, 2024).

[8] 2.1 High Level Logical Model of SNOMED CT - Release File Specification - SNOMED Confluence (ihtsdotools.org)

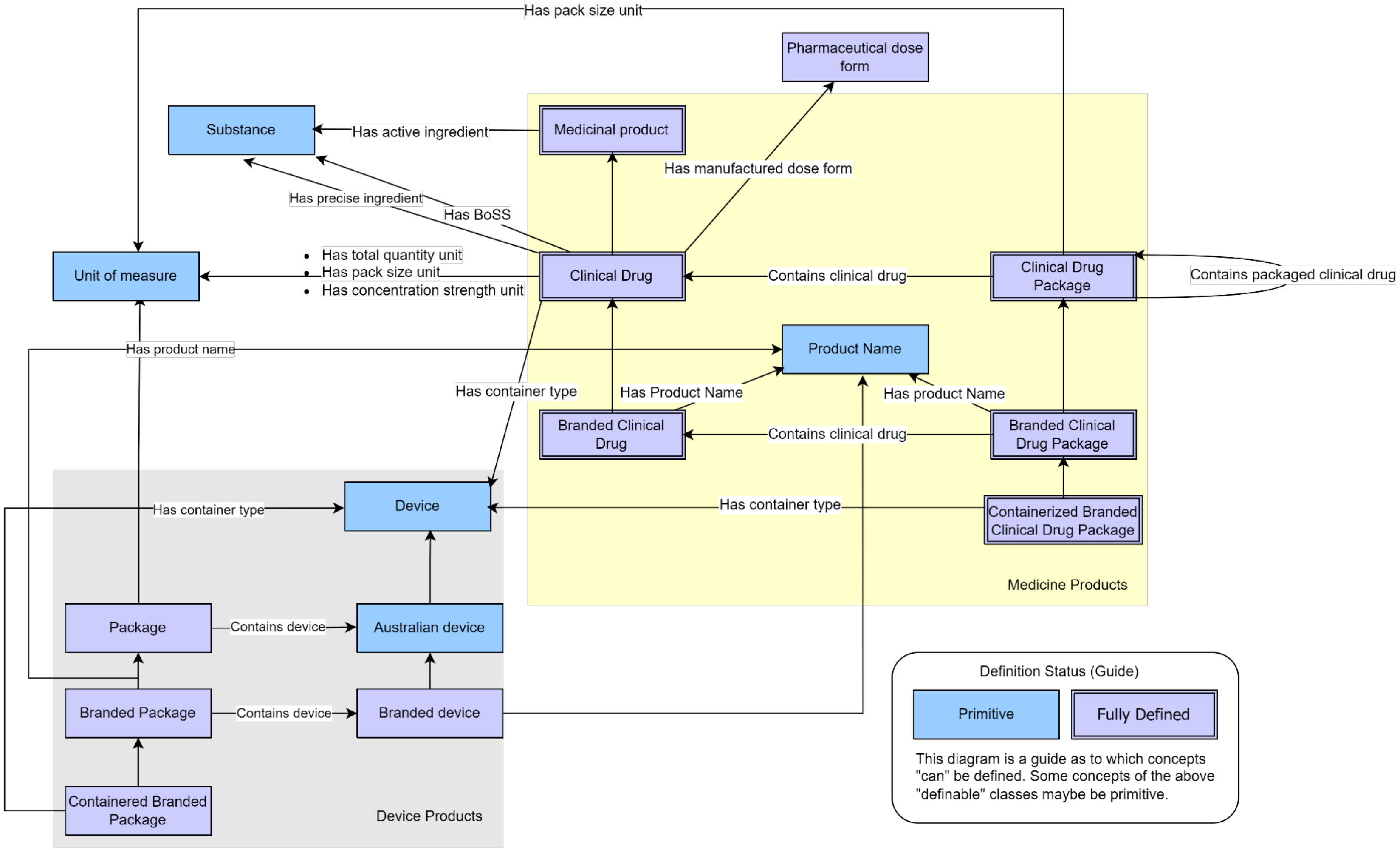


Figure 2: The AMT (v4) concept model

3.3 Seven notable reference sets

The “seven notable reference sets” are central to the AMT model. The concepts within each reference set represent different levels of abstraction of branded products and their generic product equivalents at various levels of granularity. Figure 3 includes a brief definition of the types of concepts within each of the reference sets. The specific concept types within each reference set can include those from either the 373873005| Pharmaceutical / biologic product| or 260787004|Physical object| top level hierarchies. Full definitions and more worked examples can be found in the *AMT v4 Model Specification* (Australian Digital Health Agency, 2024).

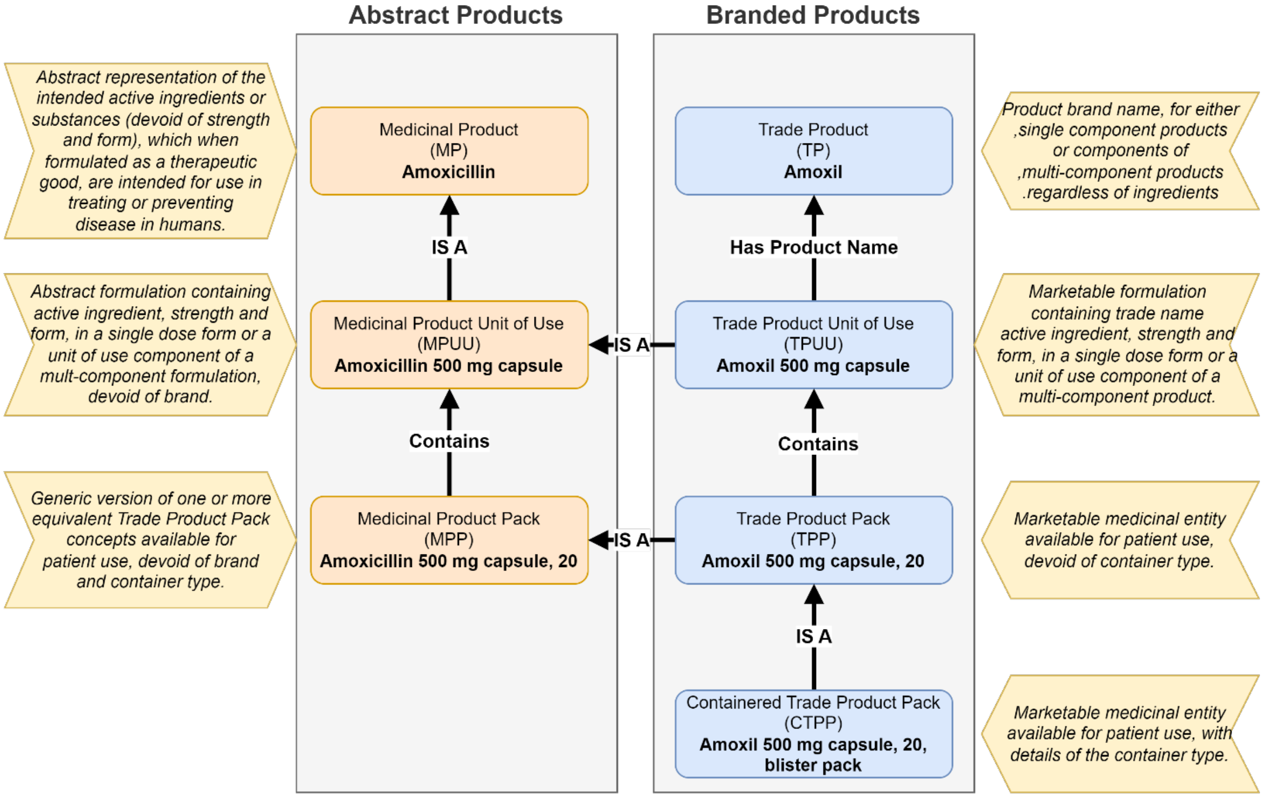

Abstract Products
Branded Products
Abstract representation of the intended active ingredients or substances (devoid of strength and form), which when formulated as a therapeutic good, are intended for use in treating or preventing disease in humans.
Medicinal Product (MP) Amoxicillin
Trade Product (TP) Amoxil
Product brand name, for either ,single component products or components of ,multi-component products .regardless of ingredients
IS A
Has Product Name
Abstract formulation containing active ingredient, strength and form, in a single dose form or a unit of use component of a mult-component formulation, devoid of brand.
Medicinal Product Unit of Use (MPUU) Amoxicillin 500 mg capsule
IS A
Trade Product Unit of Use (TPUU) Amoxil 500 mg capsule
Marketable formulation containing trade name active ingredient, strength and form, in a single dose form or a unit of use component of a multi-component product.
Contains
Contains
Generic version of one or more equivalent Trade Product Pack concepts available for patient use, devoid of brand and container type.
Medicinal Product Pack (MPP) Amoxicillin 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 the container type.

Figure 3: Seven notable reference sets within AMT and the types of concepts within each.

As mentioned above, the Seven Notable reference sets, can contain a mix of Medicine and Device concept types. Whenever the terms MP, MPUU, MPP, TP,TPUU, TPP, CTPP are used – they refer to *members of these reference sets*.

Table 2: Examples of concept types in each reference set

Reference Set	Concept types	Examples
Medicinal Product (MP)	• Medicinal Product • Device	• Sildenafil-containing product • Bandage tubular finger
Medicinal Product Unit of Use (MPUU)	• Clinical Drug • Australian Device	• Sildenafil 100 mg tablet • Bandage tubular finger bandage
Medicinal Product Pack (MPP)	• Clinical Drug Package • Package	• Sildenafil 100 mg tablet, 12 • Bandage tubular finger bandage, 1

Trade Product (TP)	- Product Name	- Ernafil - Tubegauz Refill (0501658)
Trade Product Unit of Use (TPUU)	- Branded Clinical Drug - Branded Device	- Ernafil 100 mg tablet - Tubegauz Refill (0501658) bandage
Trade Product Pack (TPP)	- Branded Clinical Drug Package - Branded Package	- Ernafil 100 mg tablet, 12 - Tubegauz Refill (0501658) bandage, 1
Containered Trade Product Pack (CTPP)	- Containered Branded Clinical Drug Package - Containered Branded Package	- Ernafil 100 mg tablet, 12, blister pack - Tubegauz Refill (0501658) bandage, 1, carton

Other concepts exist within SNOMED CT-AU “above” the MP members, and between MP and MPUU, that are not part of the 7 notable reference sets. These may be of use to some implementations, but are not currently in scope for specific AMT support and almost exclusively inherited as part of the International core content. The 7 notable reference sets identify concepts with specific levels of granularity.

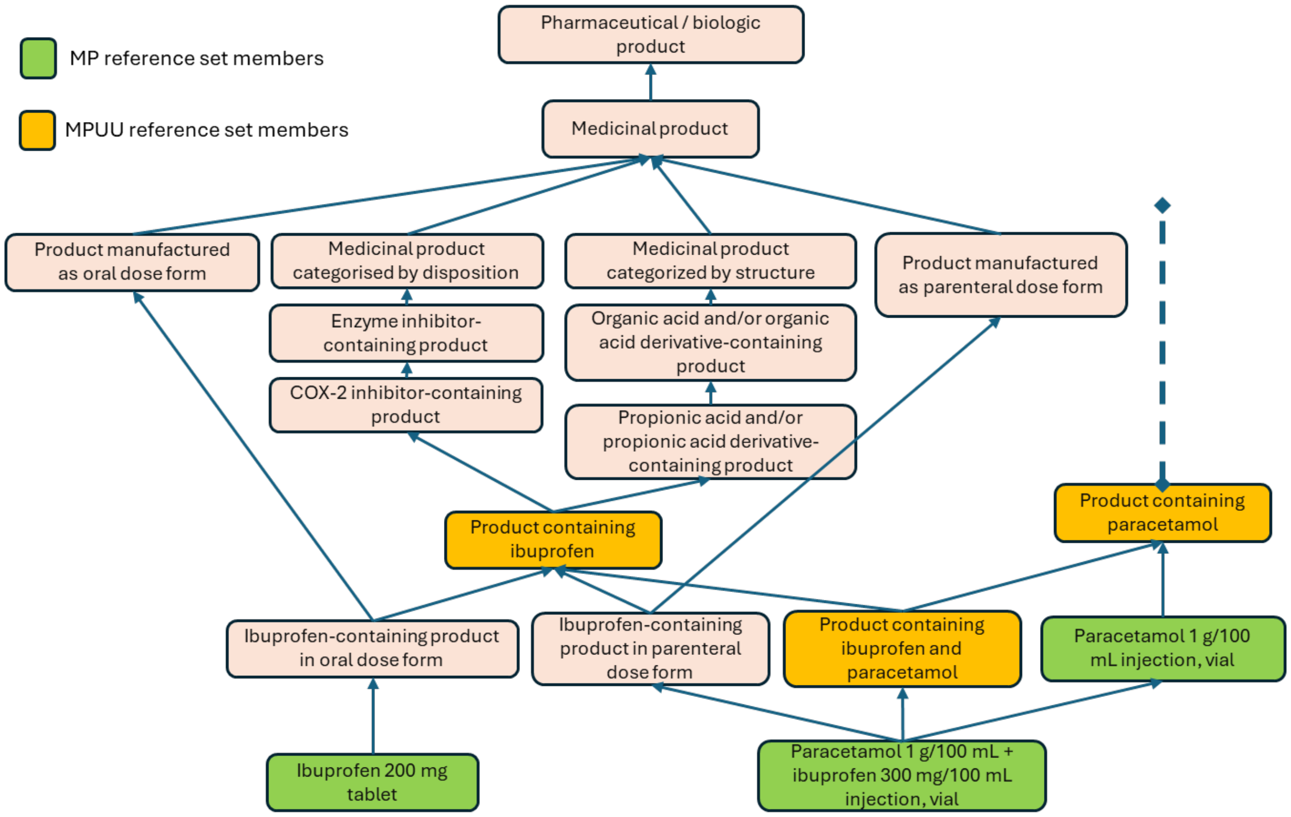

🟩 MP reference set members
🟨 MPUU reference set members
Pharmaceutical / biologic product
Medicinal product
Product manufactured as oral dose form
Medicinal product categorised by disposition
Medicinal product categorized by structure
Product manufactured as parenteral dose form
Enzyme inhibitor-containing product
Organic acid and/or organic acid derivative-containing product
COX-2 inhibitor-containing product
Propionic acid and/or propionic acid derivative-containing product
Product containing ibuprofen
Product containing paracetamol
Ibuprofen-containing product in oral dose form
Ibuprofen-containing product in parenteral dose form
Product containing ibuprofen and paracetamol
Paracetamol 1 g/100 mL injection, vial
Ibuprofen 200 mg tablet
Paracetamol 1 g/100 mL + ibuprofen 300 mg/100 mL injection, vial

Figure 4:Detailed lineage of Medicinal Product subhierarchy

4 Format of the release bundle

4.1 SNOMED CT-AU Release file structure

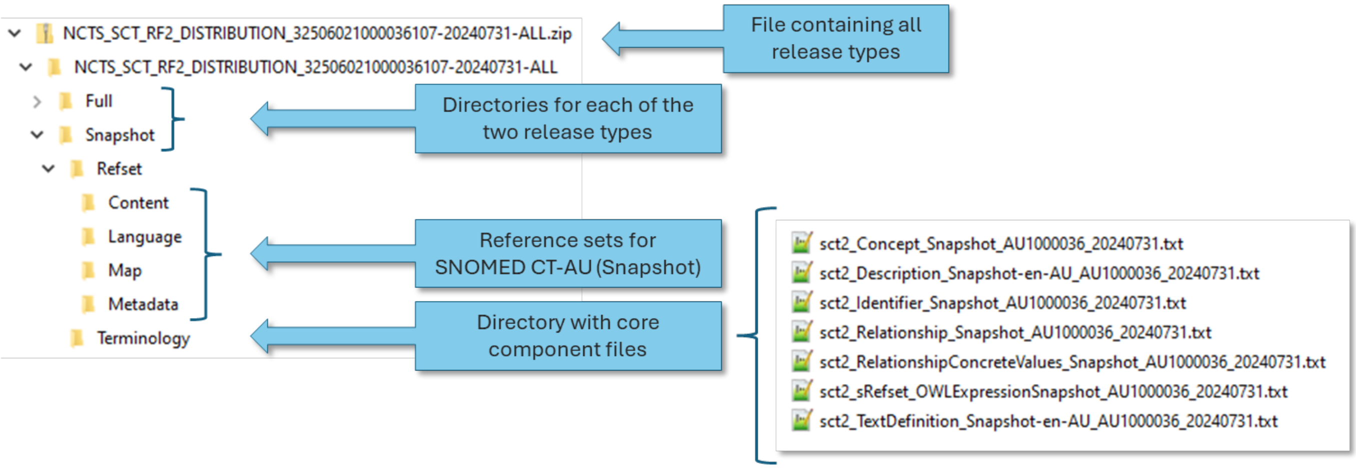
The combined SNOMED CT-AU and AMT content is distributed in a compressed file using the naming convention:

NCTS_SCT_RF2_DISTRIBUTION_32506021000036107-<releaseDate>-<releaseType>.zip

For example:

NCTS_SCT_RF2_DISTRIBUTION_32506021000036107-20170531-ALL.zip

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



```
NCTS_SCT_RF2_DISTRIBUTION_32506021000036107-20240731-ALL.zip
  NCTS_SCT_RF2_DISTRIBUTION_32506021000036107-20240731-ALL
    Full
    Snapshot
      Refset
        Content
        Language
        Map
        Metadata
      Terminology
```

```
sct2_Concept_Snapshot_AU1000036_20240731.txt
sct2_Description_Snapshot-en-AU_AU1000036_20240731.txt
sct2_Identifier_Snapshot_AU1000036_20240731.txt
sct2_Relationship_Snapshot_AU1000036_20240731.txt
sct2_RelationshipConcreteValues_Snapshot_AU1000036_20240731.txt
sct2_sRefset_OWLExpressionSnapshot_AU1000036_20240731.txt
sct2_TextDefinition_Snapshot-en-AU_AU1000036_20240731.txt
```

File containing all release types
Directories for each of the two release types
Reference sets for SNOMED CT-AU (Snapshot)
Directory with core component files

Figure 5: Directory layout of release bundle

Comprehensive details of the RF2 specification are available in *SNOMED CT Release File Specifications* (SNOMED International, 2024).

The integrity of the release bundles can be verified after downloading, using the SHA256 hash value included in the syndication feed for each package. There are many ways this can be achieved, including using tools like sha256sum[9]. We also publish a code that demonstrates how to do it from Java[10].

4.2 RF2 distribution types

The RF2 distribution type is published in two states providing different views of the terminology's history, as summarised in the following table and illustrated in Figure 6. They can be published together in a single zip as per the "ALL" example in **Error! Reference source not found.**, or separately (e.g. "SNAPSHOT.zip")

[9] Some background information can be found at https://itsfoss.com/checksum-tools-guide-linux/.

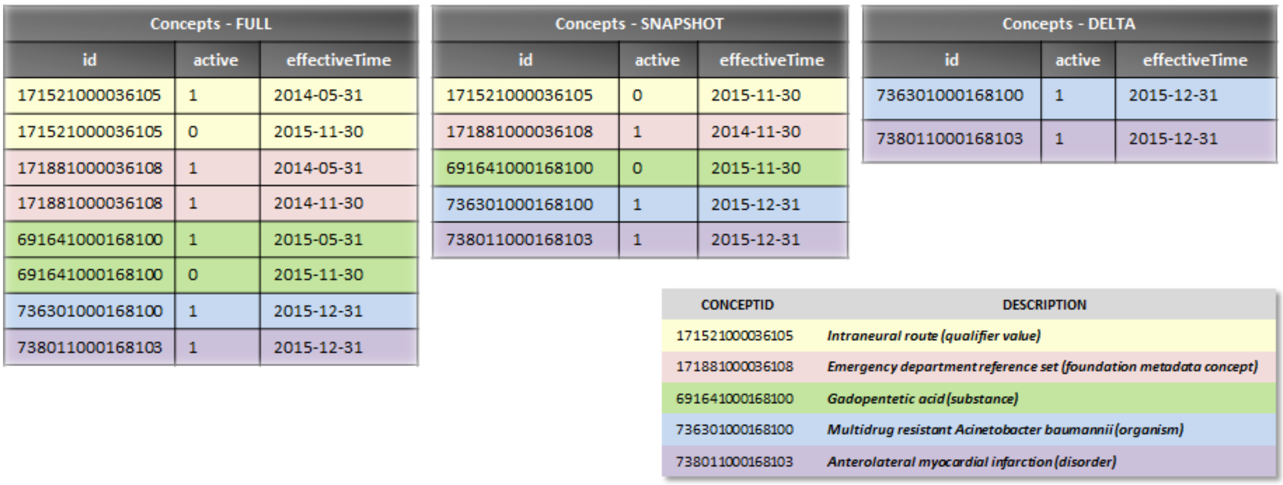
[10] Available from https://github.com/AuDigitalHealth/ncts-syndication-client.

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 only most the recent version of each component at the time of that release.
Delta (NOT PUBLISHED)	The files representing each type of component contain only component versions created **since a specific previous release**. Every entry in a Delta release represents either a new component or a change to an existing component. (Unchanged components are not included.) Deltas are always **relative** to a specific previous release.

A Delta release, showing all the changing between two specific releases can also be calculated using a command line tool published by SNOMED International[11]. Historically, Deltas were published with each release, however, with the introduction of more frequent releases by SNOMED International this was stopped to mitigate the risks of implementers missing delta. As each "published delta" is specific only to the immediately preceding release. Implementers interested in using Delta's will need to calculate their own using the available tool mentioned above.

The figure below illustrates the different release types with respect to the same set of data. The Full release contains multiple instances of some concepts, whereas the Snapshot contains one only – the most recent instance of each. Lastly, the Delta release only displays the concepts that have changed in the most recent release.



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 6: Examples of Full, Snapshot and Delta content

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

- The Snapshot release type is the easiest to query as it contains only the latest version of each component. Data updates for a Snapshot implementation typically require replacing content with a new Snapshot release.

[11] IHTSDO/delta-generator-tool: Command line Java tool to extract an RF2 delta from a particular date, from a archive containing SNOMED Full files (github.com)

- Queries written against the Full release type 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 release types 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 CT Terminology Services Guide* (SNOMED International) provides further details concerning importing the various release types.

5 Importing terminology content

Before consuming terminology codes in a clinical software system, the content needs to be loaded into a database. 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.

Exemplar scripts for creating the main tables and other related objects are available on Github[12]. These objects need to be created prior to loading content into the respective tables.

The file contains configuration scripts to set up 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.

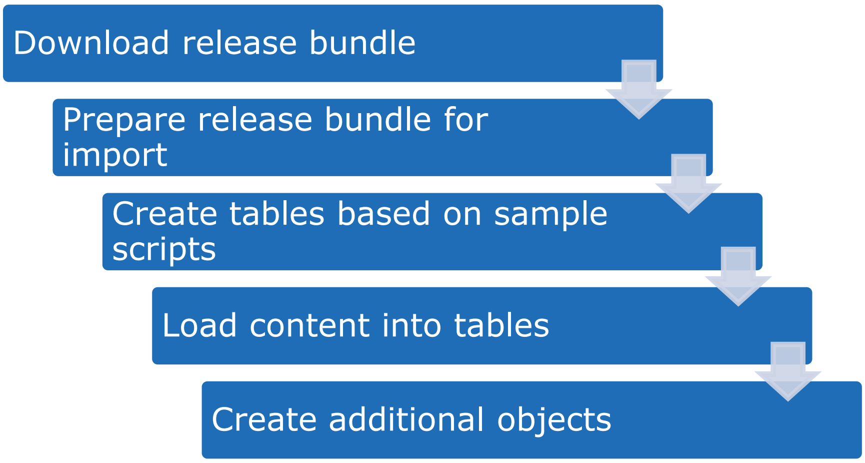

Download release bundle
Prepare release bundle for import
Create tables based on sample scripts
Load content into tables
Create additional objects

Figure 7: Process for loading terminology content

5.1 Download the release bundle

1 To download the release bundle, you must first register on the NCTS website. As part of the registration process, you will be required to accept the SNOMED CT Affiliate License Agreement, and the Australian National Terminology Licence Agreement, both of which are available at the NCTS Document Library.

2 Download the latest RF2 SNAPSHOT bundle from **NCTS website > Access > SNOMED CT-AU > Release Bundles.** Unzip the release to a suitable location.

3 Clone the Github repository containing the sample scripts on the Australian Digital Health Agency GitHub repository.

[12] AuDigitalHealth/sctau-sample-scripts: Sample relational database load scripts and SQL queries for processing SNOMED CT-AU RF2 release files. (github.com)

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.

Please note, 2 versions of the scripts are provided:

- AMTv4-sample-scripts **Current**. Only applicable for SNOMED CT-AU releases published in September 2024 onwards.
- AMTv3-sample-scripts **Deprecated**. Only for historical reference or use with releases of SNOMED CT-AU prior to September 2024.

The scripts are comparable, but not interchangeable due to changes in release format and concept model.

5.2 Create tables based on sample scripts

4 Connect to a MySQL server and create a database '**sctau'.**
(Other flavours of SQL can be used, however the syntax of queries my need to be modified accordingly)

5 Execute the **schema/1_createSchema.sql** script up a database and create common tables to store SNOMED CT-AU and AMT content.

5.3 Load content into the tables

The sample scripts contain a file to load data from the release bundle into the newly created tables. The **schema/2_populateTables.sql** file needs to be modified to replace the exemplar paths with the location of the downloaded terminology release file(5.1 above).

5.4 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 precoordinated tables to store the results of complex queries.

See Section 9 for further details on using the provided scripts.

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:

- Languages and dialects can be represented as Language type reference sets. The *Australian dialect reference set* is included in the SNOMED CTAU release.
- Developing maps to and from other code systems and classifications.
- Subsets of Concepts, Descriptions, and Relationships can be selected and presented as reference sets.

SNOMED CT-AU has a wide range of reference set content covering various health care disciplines and clinical specialties. Additionally, the methods used to create and improve existing reference sets are outlined in the *SNOMED CT-AU Development Approach for Reference Sets* (Australian Digital Health Agency, 2019).

6.1 Reference sets – types and examples

All reference sets are based on a generic data structure that

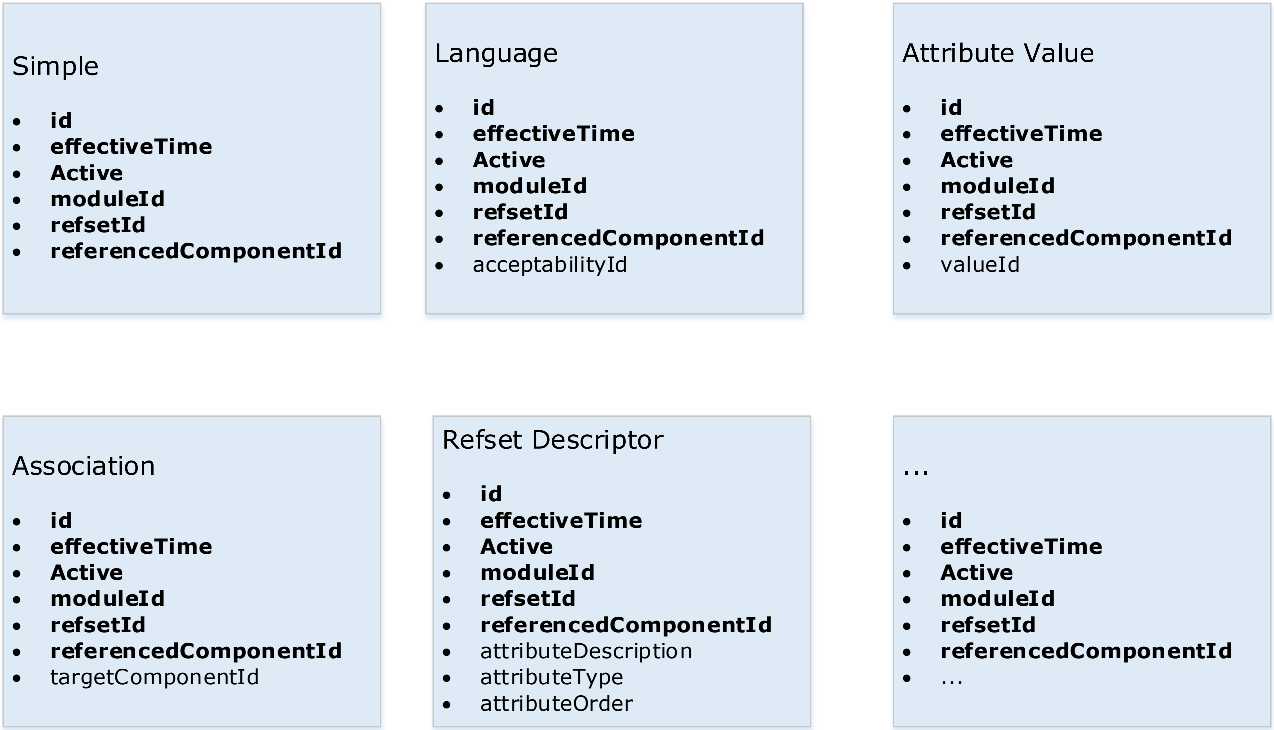
can be extended to meet application requirements. The first four fields fulfil the same purpose as their counterparts in the core Concept, Description, and Relationship files: see *SNOMED CT Release File Specifications* (SNOMED International, 2024) for related information.

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.

Field	Purpose
referencedComponentId	Uniquely identifies the component that this row relates to, thus defining membership of this component in the reference set.
(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; 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, the Agency may release reference sets of various types (patterns). This document will be extended to describe them as they are introduced; see *SNOMED CT Release File Specifications* (SNOMED International, 2024) 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 8: 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 addressed 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.

6.1.1.1 Association reference set

This pattern is used to declare some association between two concepts, for example to capture historical associations between components. This may be useful during terminology maintenance. To illustrate, the *REPLACED BY association reference set* identifies retired (inactive) components along with their active, replacement components (where a replacement exists). The **sample_queries.sql** (5.1 above) file contains a script that demonstrates how to identify inactive AMT product concepts and their replacements.

Not all inactive components will be in these reference sets. Some components may be retired without an active replacement (for example, if the concept was created erroneously) and these are not included in the reference set. An example of this is provided in the **sample_queries.sql** file.

6.1.1.2 Attribute value reference set

The *Attribute value reference set* is used to provide the details of a non-defining attribute of a component. For example, the *Concept inactivation indicator attribute value reference set* indicates the reason a concept has been inactivated. Reasons for inactivating concepts may be because the concept is erroneous, duplicated or outdated due to a change in source data.

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 the Agency 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”*
• 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 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 NCTS are categorised according to those bound to Agency 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 NCTS 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 NCTS 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 health care, 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, the seven notable reference sets have been developed for the AMT that list all concepts for each of the seven levels of granularity within the AMT concept model:

- 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 semiautomated 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, such as a reference set developed for use within a particular Agency 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 the Agency.
- 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 *SNOMED CT-AU Development Approach for Reference Sets* (Australian Digital Health Agency, 2019), 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 something like "`destinationId in (SELECT sourceId FROM transitive_closure WHERE destinationId = 56459004)`". This query would produce around 800 concepts – all musculoskeletal findings relating to foot structures.

6.3.2 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 unsuitable 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.

6.3.2.1 SQL examples

Using the example above where a reference set of pregnancy findings is needed, the requirements might be specified as including all concepts that are types of 118185001 |*Finding related to pregnancy*|. All these subtypes may be identified using the transitive closure table (described in Section 10):

```
SELECT id
FROM concepts_snapshot
WHERE id in (select distinct sourceId from transitive_closure
             where destinationId = 118185001);
```

This query produces around 1,704 concepts[13] – 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 id in (select distinct sourceId from transitive_closure
             where destinationId = 118185001)
AND id NOT IN
  (SELECT referencedComponentId
   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:

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

- 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)

Every SNOMED CT release requires an associated Language reference set to specify preferences for the different descriptions with a particular context.

SNOMED CT-AU provides two types of descriptions:

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

Language type reference sets provide a mechanism to annotate these descriptions with one of these values to provide localisation:

Preferred	Indicates that the description annotated with this value is the preferred among the descriptions of that type for that concept. Every concept should have exactly one preferred Synonym and one preferred FSN for each concept.
Acceptable	Indicates that terms annotated with this value are an acceptable (though not the preferred) way of describing a concept in the context of the Language type reference set.

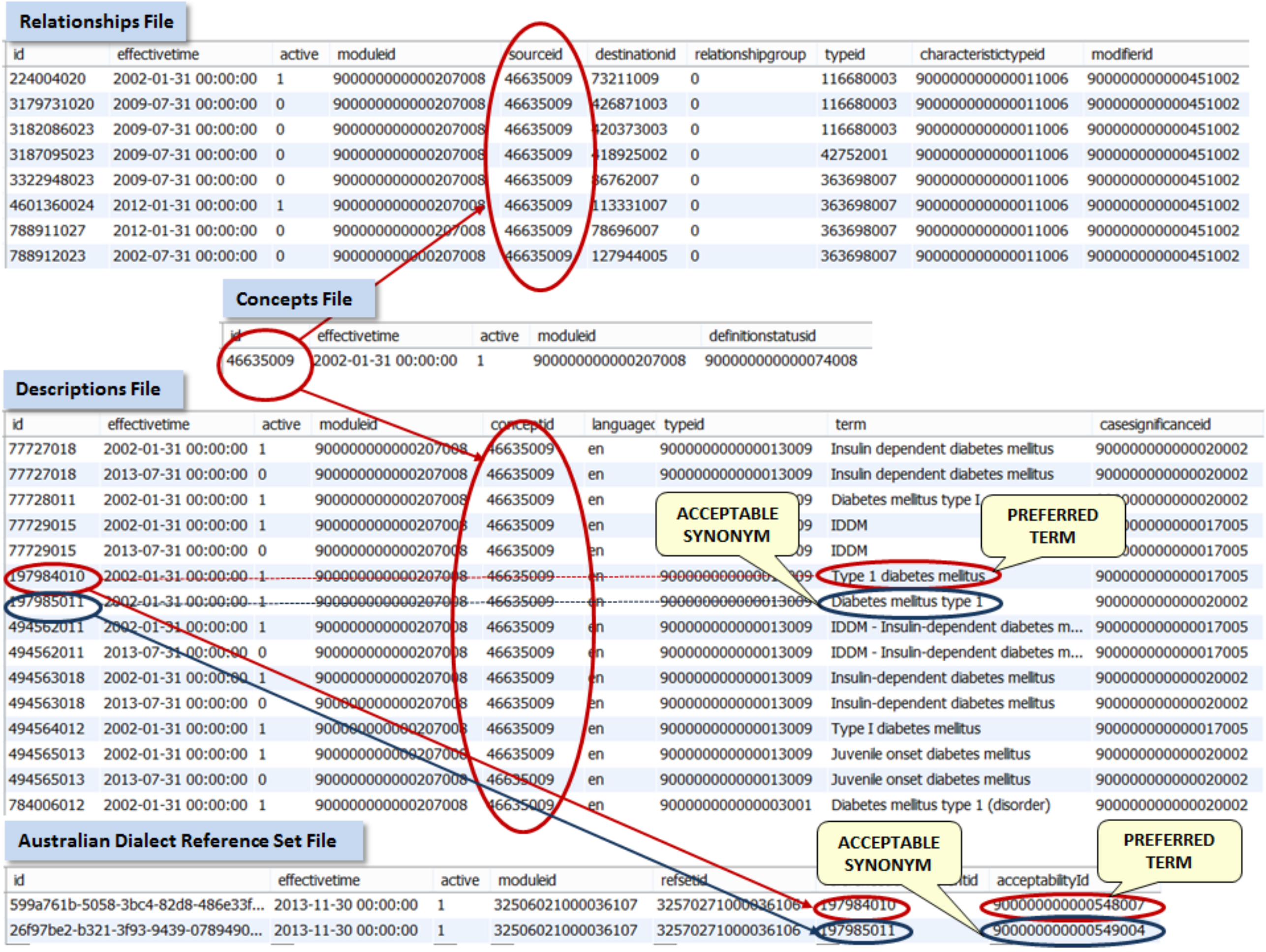
Language type reference sets do not (currently) explicitly indicate if a term is “Not Acceptable”. Instead it may be implied by the absence of a reference to those terms within the language reference set. (Though this may depend on the specific implementation).

The ADRS is the default Language type reference set for use in Australia. It specifies the descriptions identified as appropriate for the recording of clinical information in Australian e-health context. The ADRS was initially based on the UK dialect subset that is released by SNOMED International as part of the SNOMED CT International Release. Subsequent development of the ADRS continues to improve the reference set for Australian usage and will include additional Australian expressions and spellings.

Although both a preferred FSN AND Synonym are provided. It is the Preferred Synonym that should be chosen as the “Preferred Term” for use on the user interface.

Implementation of the ADRS is required, so that the Preferred Term for each concept may be identified. The ADRS is available in the **Refset > Language** folder in RF2 release bundle, for example if using Snapshot files the file path will be (where YYYYMMDD denotes the release date): SnomedCT_Release_AU1000036_YYYYMMDD\RF2Release\Snapshot\Refset\Language\der2_cRefset_LanguageSnapshot-en-AU_AU1000036_YYYYMMDD.txt

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	86762007	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	900000000000013009	Diabetes mellitus type I	900000000000020002
77729015	2002-01-31 00:00:00	1	900000000000207008	46635009	en	900000000000013009	IDDM	900000000000017005
77729015	2013-07-31 00:00:00	0	900000000000207008	46635009	en	900000000000013009	IDDM	900000000000017005
197984010	2002-01-31 00:00:00	1	900000000000207008	46635009	en	900000000000013009	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	referencedcomponentid	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

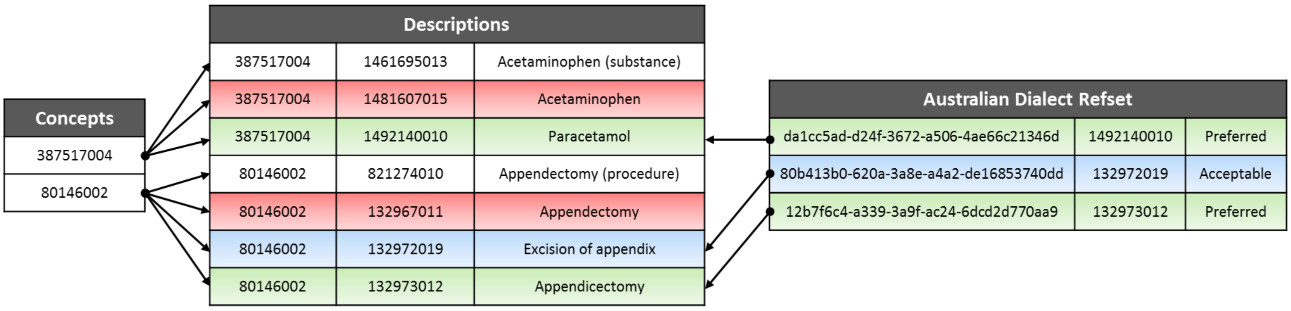
Figure 9: Retrieving the Preferred Term and acceptable Synonym for concept 46635009

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		
387517004	1461695013	Acetaminophen (substance)
387517004	1481607015	Acetaminophen
387517004	1492140010	Paracetamol
80146002	821274010	Appendectomy (procedure)
80146002	132967011	Appendectomy
80146002	132972019	Excision of appendix
80146002	132973012	Appendicectomy

Australian Dialect Refset		
da1cc5ad-d24f-3672-a506-4ae66c21346d	1492140010	Preferred
80b413b0-620a-3a8e-a4a2-de16853740dd	132972019	Acceptable
12b7f6c4-a339-3a9f-ac24-6dcd2d770aa9	132973012	Preferred

Figure 10: 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), with “Excision of appendix” as acceptable (blue), while “Appendectomy” is not acceptable.

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. Also note that columns have been omitted from the example for brevity.

Note: There is only ever a single FSN for a given language. So although Fully Specified Names are also given a preference within language reference sets, “Preferred FSN” is somewhat redundant within Australia. FSN preferences are most relevant to multilingual editions, in jurisdictions with more than one official language.

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 D.typeId = 900000000000013009 -- Synonym
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(2048)

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 D.typeId = 900000000000013009 -- Synonym
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 value relationships

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.

Typical relationships in SNOMED CT allow a concept to have an attribute with a concept value (e.g. "21433011000036107 |*paracetamol*|"), whereas data type properties provide attributes that have a concrete data value (e.g. “5”). Data type properties are modelled using Concrete value relationships, as described by the RF2 Specification[14].

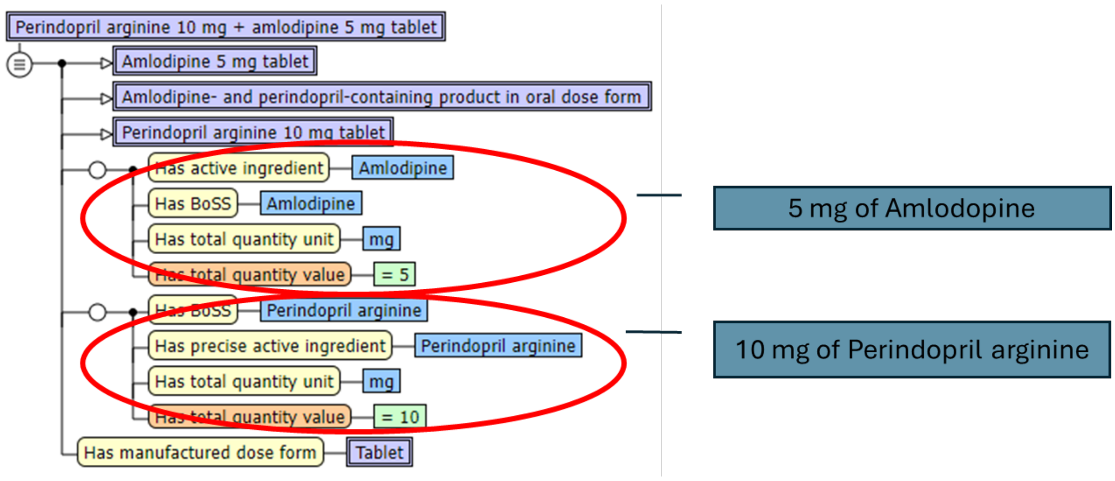
The AMT model uses the concrete domains of real numbers, integers and strings. These are used to represent the defining numeric attributes of AMT concepts, specifically:

- *Strength* – e.g. the concentration of an ingredient, or total amount in a discrete unit.
- *Pack size* – e.g. the volume of a medicine in an ampoule, or the number of tablets in a pack.
- *Subpack quantity* – e.g. the number of containered components within pack.
- *Property counts – used to ensure the terminology accurately represents the “closed word”*[15] *relationships between similar products*

String values are used to specify *other identifying information* – Defining information, that is not supported by the general AMT Concept Model, but important to differentiate products.

The Concrete relationship file is identical to the standard relationship file, however the destinationId field is replaced by (concrete) value. Most concrete domain properties are role grouped such that they are associated with other properties – e.g. ingredient and units to specify strength.

Figure 11: Example of concrete domains role grouped with other properties



Role grouping is particularly important when some properties appear more than once on a concept (e.g. more than one ingredient) to ensure the correct properties are associated with each other. Relationships and Concrete value relationships in the same role group, have the same relationshipGroup value in the distribution file. Note these relationshipGroup values are concept specific (not unique across the terminology).

6.5.1 Using concrete value relationships – Strength example

Concrete values allow queries such as “Products with more than 500 mg of Amoxicillin in a single unit” to be made against the terminology

[14] https://confluence.ihtsdotools.org/display/DOCRELFMT/4.2.6+Concrete+Value+File+Specification

[15] See: D.3 for more information.

Terminology server

Terminology servers provide the most straightforward mechanism to perform such queries. Using Expression Constraint Language (ECL) we can compare these two queries

1. MPUUs containing Amoxicillin
   ```
   ^929360071000036103|MPUU|:732943007|Has BoSS|=372687004|Amoxicillin|
   ```
2. MPUUs containing more than 500mg of Amoxicillin total.

```
^929360071000036103|MPUU|:
{732943007|Has BoSS |=372687004|Amoxicillin|,
 999000041000168106|Has total Qty|>#500,
 999000051000168108|Has total Qty unit|=258684004|mg|}
```

Query 1 (All Amoxicillin MPUUs)	Query 2 (Amoxicillin > 500mg)
Amoxicillin 1 g injection, vial	Amoxicillin 1 g + clavulanic acid 200 mg injection, 1.2 g vial
Amoxicillin 100 mg/mL powder for oral liquid	Amoxicillin 1 g injection, vial
Amoxicillin 2 g + clavulanic acid 200 mg injection, 2.2 g vial	Amoxicillin 1 g tablet
Amoxicillin 250 mg capsule	Amoxicillin 2 g + clavulanic acid 200 mg injection, 2.2 g vial
Amoxicillin 3 g powder for oral liquid, sachet	Amoxicillin 3 g powder for oral liquid, sachet
Amoxicillin 500 mg + clavulanic acid 100 mg injection, 600 mg vial	Amoxicillin 875 mg + clavulanic acid 125 mg tablet
+ 13 more	

SQL Sample Script

Comparable queries producing the same result as above using the SQL Sample scripts (assuming all entries are active for demonstration brevity) are:

1 MPUUs containing Amoxicillin

```
SELECT referencedComponentId, get_PT(referencedComponentId)
FROM refset_snapshot MPUU
JOIN relationships_snapshot AS Ingredient
ON MPUU.referencedcomponentid = Ingredient.sourceid
AND MPUU.refsetId = 929360071000036103 AND Ingredient.typeid = 732943007
WHERE Ingredient.destinationid = 372687004;
```

2 MPUUs containing more than 500mg of Amoxicillin total.

```
SELECT referencedComponentId, get_PT(referencedComponentId)
FROM refset_snapshot MPUU
JOIN relationships_snapshot AS Ingredient
ON MPUU.referencedcomponentid = Ingredient.sourceid
AND MPUU.refsetId = 929360071000036103 AND Ingredient.typeid = 732943007
AND Ingredient.destinationid = 372687004
JOIN relationships_concrete_values_snapshot AS strengthValue
ON Ingredient.sourceId = strengthValue.sourceid AND
Ingredient.relationshipgroup = strengthValue.relationshipgroup AND
strengthValue.typeid = 999000041000168106
JOIN relationships_snapshot AS StrengthUnits
ON strengthValue.sourceId = StrengthUnits.sourceid AND
strengthValue.relationshipgroup = StrengthUnits.relationshipgroup AND
StrengthUnits.typeid = 999000051000168108
```

```
    WHERE strengthValue.value > 500
    AND StrengthUnits.destinationid = 258684004;
```

Alternatively, using the denormalised table produced with the sample scripts

```
SELECT distinct mpuuid, mpuuterm from v4_ingredient_strength
WHERE bossid = 372687004
AND TotalQuantity > 500
AND TotalQuantityUnitId = 258684004;
```

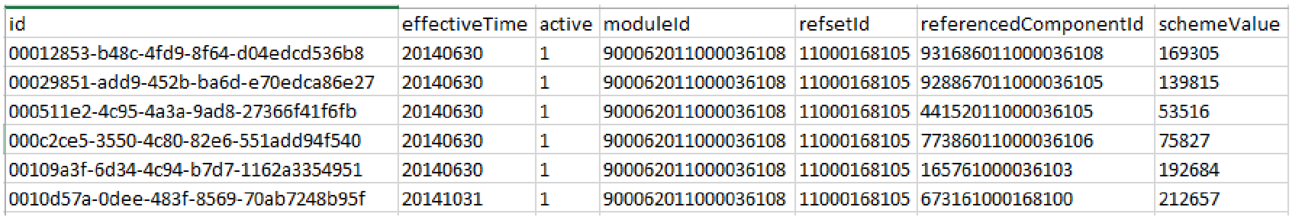
Please see 8.4.4 for further information and considerations around strength.

6.6 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 provided as a simple map to concepts in the CTPP reference set.

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 12: 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 pressurised inhalation, 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. A broad overview of implementation processes is available in the *SNOMED CT-AU Clinical Terminology Implementation Process and Checklist* (Australian Digital Health Agency, 2017).

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 based systems.

In this case, the implementer may prefer to use SNOMED CT-AU 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. To achieve this, SNOMED CT-AU reference sets can be used as a basis for mapping the local terminology codes to SNOMED CT-AU, and vice versa.

To send a message, the local codes need to be mapped to a SNOMED CT-AU code. Where an agreed set of SNOMED CT-AU codes are 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 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:

- *NCTS Guidance for People and Processes* (Australian Digital Health Agency, 2018)
- *NCTS Guidance for Use in Healthcare Software* (Australian Digital Health Agency, 2018)
- *NCTS Use of Medical Nomenclatures in Information Exchange* (Australian Digital Health Agency, 2018)
- *SNOMED CT-AU Mapping Guidelines* (Australian Digital Health Agency, 2022)

7.2 Native implementation

An alternative to mapping Australian terminology to a local or proprietary code set is to directly implement the Australian terminology – such that the system uses it “natively”.

One of the first technical considerations when implementing the 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 SNOMED CT-AU (including AMT) natively can be found in the following documents:

- *NCTS Guidance for People and Processes* (Australian Digital Health Agency, 2018)
- *NCTS Guidance for Use in Healthcare Software* (Australian Digital Health Agency, 2018)

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. However, all may be useful to assist with search experience.

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 as the native coding system when new systems are developed, enabling easier interoperability. To implement SNOMED CT-AU 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 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 Bespoke implementations versus terminology servers

These two approaches have their distinct advantages and disadvantages, so it is up to each implementer to determine which set will be the most compatible with their needs. As a general rule though, 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.

7.3.1 Bespoke implementations

Implementations with bespoke terminology data structures and implementation can provide fast functionality, well targeted to the system use cases. However, potential disadvantages of this approach include:

- 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 content updates

- Possible changes to concept models and distribution formats, may require refactoring of the implementation.
- Reduced ability to centrally manage terminology for multiple systems in an organisation.
- Complex reasoning features provided by terminology servers are hard to reproduce cheaply.

7.3.2 Terminology servers

Terminology servers provide an alternative which, depending upon the product chosen, can mitigate or eliminate most of the disadvantages of bespoke implementations. However, the potential pitfalls of this approach include:

External point of failure: An external terminology server, depending upon the nature of the deployment, how it is used within an application[16], 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.

[16] That is, real-time access or offline with updates.

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 6.1 “Text Searches” of the *SNOMED CT Terminology Services Guide* (SNOMED International) 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 may not be used, and performance may 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 is an 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* (SNOMED International, 2024) 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 NCTS 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.

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 pre-created 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[17], 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.

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:

1. The terminology concept identifier of the code selected. For example:
    - If using SNOMED CTAU natively, this could be the SNOMED CT identifier of an AMT concept like 91935009.
    - If using a mapped implementation, this will be the code from the local medicines dictionary for the selected item.

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

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

- If using SNOMED CTAU 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.

3 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 versions of the SNOMED CT-AU:

```
"http://snomed.info/sct/32506021000036107/version/{effectiveTime}"
```

The effectiveTime value is the release date specified for the terminology release files. For example, a component released in the January 2025 Australian release will have an effectiveTime value of “20250131”.

```
"http://snomed.info/sct/32506021000036107/version/20250131"
```

Examples of different encodings of the version of the SNOMED CT-AU January 2025 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/20250131"
displayName="Lorano 10 mg uncoated tablet, 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/20250131" />
     <code value="33256011000036105" />
     <display value="Lorano 10 mg uncoated tablet, 30” />
</xyz>
```

8.3.3 JSON

```
"xyz": [
   {
      "system": "http://snomed.info/sct",
      "version": "http://snomed.info/sct/32506021000036107/version/20250131",
      "code": "33256011000036105",
      "display": "Lorano 10 mg uncoated tablet, 30"
   }
]
```

For further information on the URI standard that governs the application of this versioning, see the *SNOMED CT URI Standard* (SNOMED International).

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 95% of active Preferred Terms for AMT are less than 83 characters. The graph below shows the distribution of active Preferred Term lengths as of May 2024.

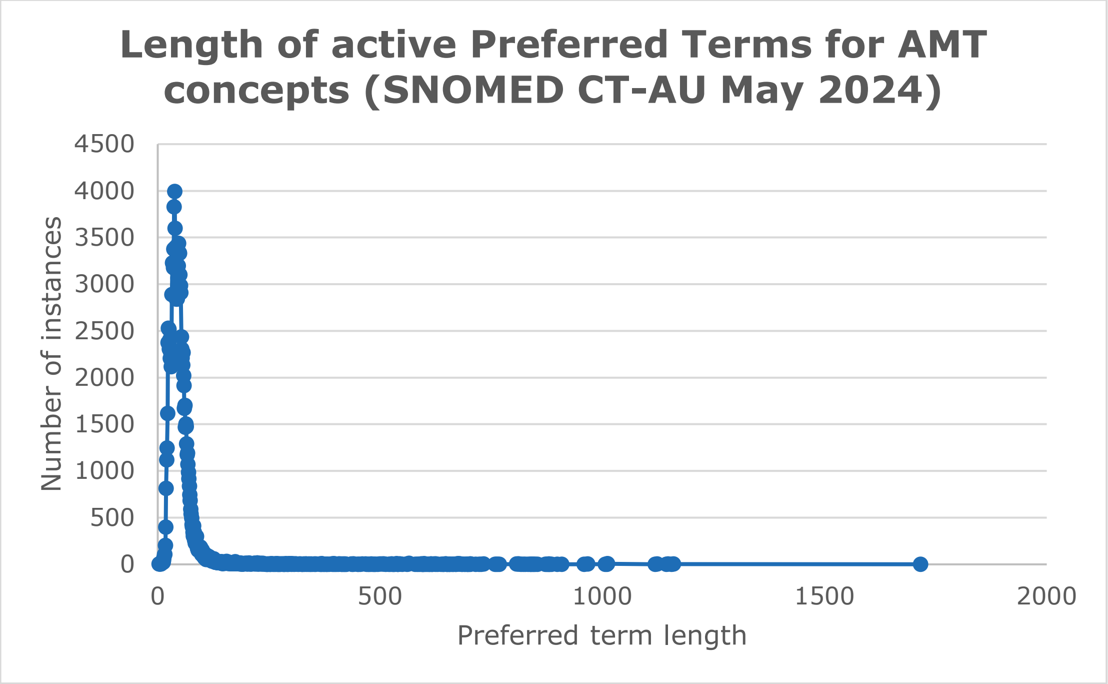


Figure 13: Active Preferred Term lengths of the AMT product concepts in the SNOMED CT-AU March 2019 production release

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 NCTS for implementation guidance and clinical safety considerations.

The SNOMED CT-AU and AMT editorial rules are subject to ongoing revaluation in response to stakeholder needs.

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 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 AMT FAQs page.

8.4.4 Strength considerations

8.4.4.1 Sufficiency of floating-point strength accuracy

Concentration representations of strength are normalised to denominator of 1, to enable more direct comparison across related products. For example, *Amikacin 1 g/4 mL injection, vial* and *Amikacin 500 mg/2 mL injection, vial* both have the same concentration (250mg/ml).

When normalising values to populate the value *Concrete value relationship*, the result includes some real numbers that repeat after the decimal point (for example, 0.333333333...). It was determined that precision to 6 decimal places sufficiently accurate strength value when calculating the non-normalised strength attribute, such as 16666.666667, 33333.333333 and 149.253731.

For example, the strength value of the MPUU *epoetin beta 5000 units/0.3 mL injection, syringe* is 16666.666667. This product has a *Unit of use size* of 0.3 mL, thus recalculating non-normalised (human-readable) strength results in “5000.0000001 international units/0.3 mL” which affords sufficient accuracy. And when rounding is applied, produces the original number.

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.

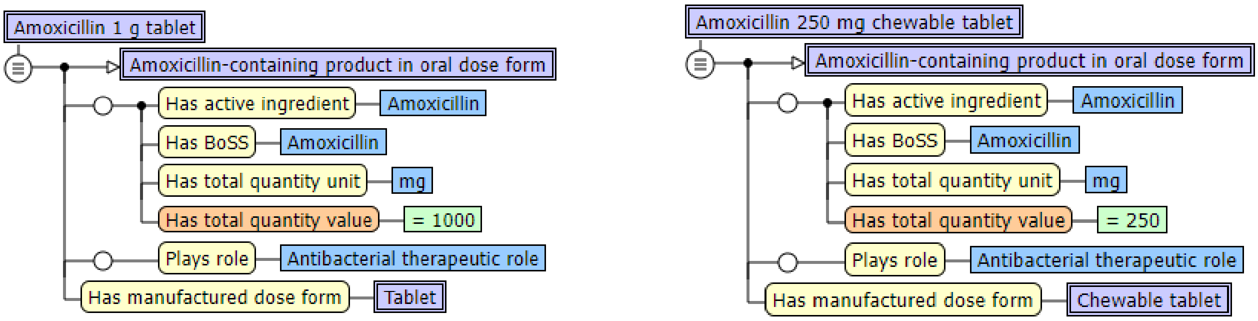
Table 6: Sample MPUU floating point strength values

MPUU ID	MPUU Preferred Term	BoSS	has concentration value	has concentration unit
21996011000036108	Epoetin beta 5000 units/0.3 mL injection, syringe	Epoetin beta	16666.666667	international unit/mL
22082011000036102	Follitropin beta 900 units/1.08 mL injection, cartridge	Follitropin beta	833.333333	international unit/mL
23315011000036101	Anakinra 100 mg/0.67 mL injection, syringe	Anakinra	149.253731	mg/mL

8.4.4.1 Normalised mass and volume units of measure

All mass and volume units of measure across AMT are normalised to milligram and millilitre respectively. For most products this is consistent with the units used in the term, but it can differ based on editorial naming conventions. Providing a consistent and predictable representation provides a reliable target for calculations involving concrete domain, and unit conversions.

Figure 14: Two products with different units in term, but same in modelling



Other (non-mass/volume) units of measure will be used as per registration. E.g. ELISA unit, kBq, mmol, Trillion vector genomes.

Implementers who are interested in using the atomically accessible strength details should be mindful of the units of measure especially if a 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).

Unit conversion factors are not currently included in the AMT data.

8.4.4.2 Unit of measure of patches

Certain products with a *Form* of “patch” use the “has concentration strength” to represent a release rate. 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. Instead, units of measure with the relevant time frame are used - “mg/24 hours” “mg/16 hours” etc.

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*|.

8.4.4.3 Dose-based prescribing ingredient selection

When using the AMT for dose-based prescribing (that is, where an ingredient and strength, rather than a specific product, is chosen) it is recommended that substances participating in 732943007|Has BoSS| relationships be used. The concept IDs for these Basis of Strength Substance (BoSS) ingredients can be determined from any release as:

```
select distinct destinationId
from relationships_snapshot
where active = 1 and typeid = 732943007;
```

The Preferred Terms for these concepts can be found using an approach such as that described in Section 9.4.3.

It is important that users are given the choice to select the BoSS ingredient because the strength of AMT products is determined by this. Similar products can have different strengths depending on the actual compound within the product. For example:

Perindopril – This medicine is available as either arginine or erbumine salts, while sharing a common MP:

- 129486004|*Perindopril*|

- 21967011000036108 |*Perindopril arginine 10 mg tablet*|
- 22192011000036107 |*Perindopril erbumine 8 mg tablet*|

In this instance, an ambiguous order is produced if just the base substance *perindopril* was selected with a strength of 10 mg. A direct interpretation of this order results in a larger dose than is provided by either of the MPUU concepts.

Pseudoephedrine – This medicine is available in both hydrochloride and sulfate compounds, with strengths relative to the specific compound.

- 62071011000036109 |*pseudoephedrine hydrochloride 120 mg tablet*|
- 86233011000036101 |*pseudoephedrine sulfate 120 mg tablet*|

Both of these products are 120 mg tablets but have different amounts of actual pseudoephedrine present. Prescribing against the BoSS reduces the likelihood of miscommunication about the desired strength.

8.4.4.4 Total Quantity and Concentration

There are two ways to describe medicine "Strengths" within AMT, depending on the specific type of product.

- Discrete (countable) forms (such as tablets and capsules) will have a "total quantity".
e.g. "*Doxycycline 100 mg tablet*" – the total quantity of doxycycline in the tablet is 100 mg.
- Continuant forms (such as liquids, creams and powders) will have a "concentration".

  e.g. "*Paracetamol 48 mg/mL oral liquid*" the concentration of paracetamol is 48 mg/mL.

Some continuant form MPUUs are containered.

- If the "pack size" and known – both a concentration and total quantity may be provided.
e.g. "*Gentamicin 80 mg / 2 mL injection, 2 mL ampoule (clinical drug)*"
Contains 2mL of a solution of 40mg/mL Gentamicin. For a total quantity of 80mg of Gentamicin.
- If containered and the "pack size" is unknown only a total quantity will be provided (as per TGA registration).

Medicinal concept Preferred Terms may still use the most preferred human representation; however, the concentration modelling will always use normalised values with 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".

The Preferred Term of some products describes only an alternate strength representation such as "1%" for 65194011000036108 |*Tinaderm 1% cream*|. The alternate strength representation is not currently published as part of the AMT data.

8.4.4.5 Products with no BoSS strength

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

- Foods/nutritional supplements (such as vitamins, minerals and trace elements with carbohydrate).
- Inert substances, diluents.

Most AMT products in the Physical object, hierarchy will not have a strength either:

- Diagnostic strips (for example, glucose indicator blood).
- Non-medicated dressings/bandages (such as bandage tubular; bandage retention cohesive heavy).

8.4.5 Restricting queries to products with a single ingredient

Scenarios in which dose-based prescribing is used are usually concerned with single (active) ingredient products. In such circumstances, it might be necessary to restrict queries to exclude those that have other active ingredients. One way to do this is count the number of HAS ACTIVE INGREDIENT or HAS PRECISE INGREDIENT relationships for each product concept, and only return those with 1.

```
select sourceId,get_PT(sourceId) from relationships_snapshot
where typeId IN (127489000,762949000)
AND moduleId = 32506021000036107
group by sourceId
having count(destinationId) = 1;
```

Alternatively, a direct “ingredient count” property is available, on TPUU concepts.

```
SELECT * FROM relationships_concrete_values_snapshot
WHERE typeId = 1142140007
AND moduleId = 32506021000036107 -- ingredient count
AND VALUE = 1;
```

8.4.6 Finding the proximal MP for an MPUU

An important characteristic of the relationships file is that it provides implementers with an inferred view of a concept definition that includes only defining relationships that are necessarily true (i.e. necessary normal form). In practical terms, this means that when a description logic classifier is run over the data prior to release, the software examines all the authored IS A and attribute relationships, finds those that are redundant and drops them, and infers additional relationships between concepts. Classifiers allow those who author and distribute terminology to manage the evolution of the content and publish an efficient release bundle.

With the introduction of the AMTv4 model, the implications when traversing the AMT to find the MP for a given MPUU, became further complex.

Since the relationships table will not contain all possible IS A relationships, a script is provided to generate all IS A relationships in a table called transitive closure and enable subsumption queries. See Section 10 and Appendix A for more details.

Additional conditions are then necessary to find the proximal MP for a given MPUU because

- Some MPs are a subtype of other MPs due to multiple ingredients, e.g. 425741009|*Buprenorphine + naloxone*| IS A 31684002|*Buprenorphine*| and IS A 89018006|*Naloxone*|, hence some MPUUs will have more than one parent MP as a result.

- Additional concepts that are not part of the “notable AMT reference sets” may be present between an MPUU and MP.

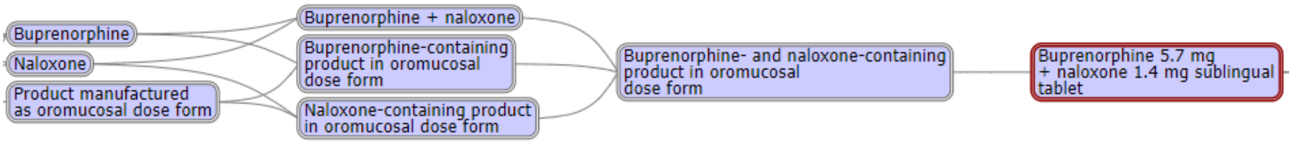

Buprenorphine
Naloxone
Product manufactured as oromucosal dose form
Buprenorphine + naloxone
Buprenorphine-containing product in oromucosal dose form
Naloxone-containing product in oromucosal dose form
Buprenorphine- and naloxone-containing product in oromucosal dose form
Buprenorphine 5.7 mg + naloxone 1.4 mg sublingual tablet

Figure 15: Some multi-ingredient MPs will have parents to the single ingredient MP if these exist within the terminology, as well as other concepts that are not part of the AMT 7 box model.

One method to resolve this is to compare the intended active ingredients and match the MPUU with the exact same ingredients as the MP, no more and no less. This is because MPs are defined by their intended active ingredients, and an MPUU will always inherit the attributes from its parent MP, so we can make use of this relationship in both directions. A script in the **AMT_use-cases.sql** file[18] demonstrates how this can be achieved.

Note that inferences are created from the meaning of the concepts that exist in the data – intermediary concepts are not authored in order to have a consistent number of steps in a pathway (traversal). For example, 87210011000036103 |*Nicotine 2 mg sublingual tablet*| has a direct relationship to MP 323283001|*Nicotine*| because “nicotine 2 mg tablet” does not currently exist in the data.

8.4.7 Denormalised AMT product tables

Implementers of the AMT, especially those who based their database structure around a previous version of the AMT (AMTv2), may store denormalised data structures to simplify and improve query performance. Section 9.5.2 describes some useful precoordinated tables provided in the SNOMED CT-AU sample scripts.

One of these is to line up the seven product concepts together in one table, with preferred terms to allow for human-readable manual validation. This requires the use of the transitive closure, relationship and reference set tables several times, with the descriptions table and Australian dialect reference set. The script is available in the **6_createAMTObjects.sql** file**[19]**. Some notes about the results set:

- Each line represents a unique combination of CTPP, ARTG ID, TPP, TPUU, TP (from TPUU), TP (from TPP), MPP, MPUU, MP concepts. This means that for any given product concept type, there can appear to be duplicates when a column is used in isolation.
- ARTG ID is included because many users find this to be a useful foreign key. However, there are empty (blank) cells included in the results set because not everything within the AMT is also listed in the ARTG. See Section 6.6 for more information about the *ARTG ID reference set*.
- Two results for TP concepts are included, where the source is the TPP and the TPUU. This allows for the scenario where a product is marketed both on its own and as part of a combination treatment, each going by a different trade name.
- Only active concepts are represented in the results.
- The function used to derive the preferred term is described in Section 6.4.1.1.

[18] Available from https://github.com/AuDigitalHealth/sctau-sample-scripts.
[19] Available from https://github.com/AuDigitalHealth/sctau-sample-scripts.

In addition, we have published an alternative way to generate this table using Java code for implementers who prefer this method.[20]

Please contact us via help@digitalhealth.gov.au for assistance with customising the results set further.

8.5 Clinical safety

To support the clinically safe use of SNOMED CT-AU or AMT in clinical information systems, system developers, vendors and healthcare organisations need to:

- Ensure that they utilise standards and industry best practice when developing and implementing their systems.
- Conduct clinical safety activities to effectively assess the clinical risks during their testing and deployment phases.
- Ensure that effective implementation of controls is in place to reduce clinical risk.

Note: If the concept identifier is not available, local codes in a system should suffice to trigger any relevant decision support alerts where required.

8.6 Product availability

Initial inclusion of a medicine product in AMT is based on its registration by a sponsor with the TGA's ARTG. However, this does not mean the product is available in the Australian supply chain.

Equally, when a product is removed from the supply chain and no longer available for sale, the corresponding AMT concepts are not retired or deprecated (that is, they remain as active). This is because AMT is a terminology describing known concepts, not a product database. The concepts are still needed to support existing health records (for example, past medical history) or e-health messages regardless of the current availability of the product.

Product availability information must be sourced outside of the AMT.

8.7 Parsing descriptions

The descriptions contained in the AMT are structured according to editorial rules, as defined in the *AMT Editorial Rules* (Australian Digital Health Agency, 2020).

While AMT's descriptions are very structured, they are not intended to be parsed into smaller components. Parsing AMT descriptions presents risk and is strongly discouraged. Required atomic data should be sourced from the appropriate source within the AMT model. For examples of extracting ingredient strengths, please refer to Section 9.5.2.2.

8.8 Modifying or extending SNOMED CT-AU and AMT

8.8.1 Modifying SNOMED CT-AU and the AMT

SNOMED CT content may not be modified. This includes SNOMED CT-AU and AMT concepts, descriptions, relationships and reference sets.

The integrity of SNOMED CT-AU and AMT must be maintained as distributed when it is implemented into local systems. That is, relationships between concepts, codes and descriptions will not be edited or distorted in use.

[20] Available from https://github.com/AuDigitalHealth/amt-flat-file-generator.

8.8.2 Extending SNOMED CT-AU and AMT

Implementers may extend SNOMED CT-AU and AMT in the following ways:

- Developing custom reference sets for specific purposes, see *SNOMED CT-AU Development Approach for Reference Sets* (Australian Digital Health Agency, 2019).
- Creating customised terminology descriptions for SNOMED CT-AU and AMT concepts, see *NCTS Guidance for People and Processes* (Australian Digital Health Agency, 2018).
- Creating locally defined concepts, see *NCTS Guidance for People and Processes* (Australian Digital Health Agency, 2018).

It is recommended that users wishing to create extensions to SNOMED CT-AU and AMT contact the NCTS for guidance and assistance.

9 Configuration scripts and sample SQL Queries

This section outlines the content and purpose of the SNOMED CT-AU sample scripts file.[21]

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 7: 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 precoordinated tables to store the results of complex queries.
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.

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.

[21] Available from https://github.com/AuDigitalHealth/sctau-sample-scripts.

Table 8: 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.
Concrete value relationships	relationships_concrete_values_snapshot	Relationships for which the target is a concrete value. (Number or String).
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, both published and transitively inferred.

9.1.1 Entity-relationship diagram

The diagram below shows the schema design for the core SNOMED CT tables as used within these sample scripts.

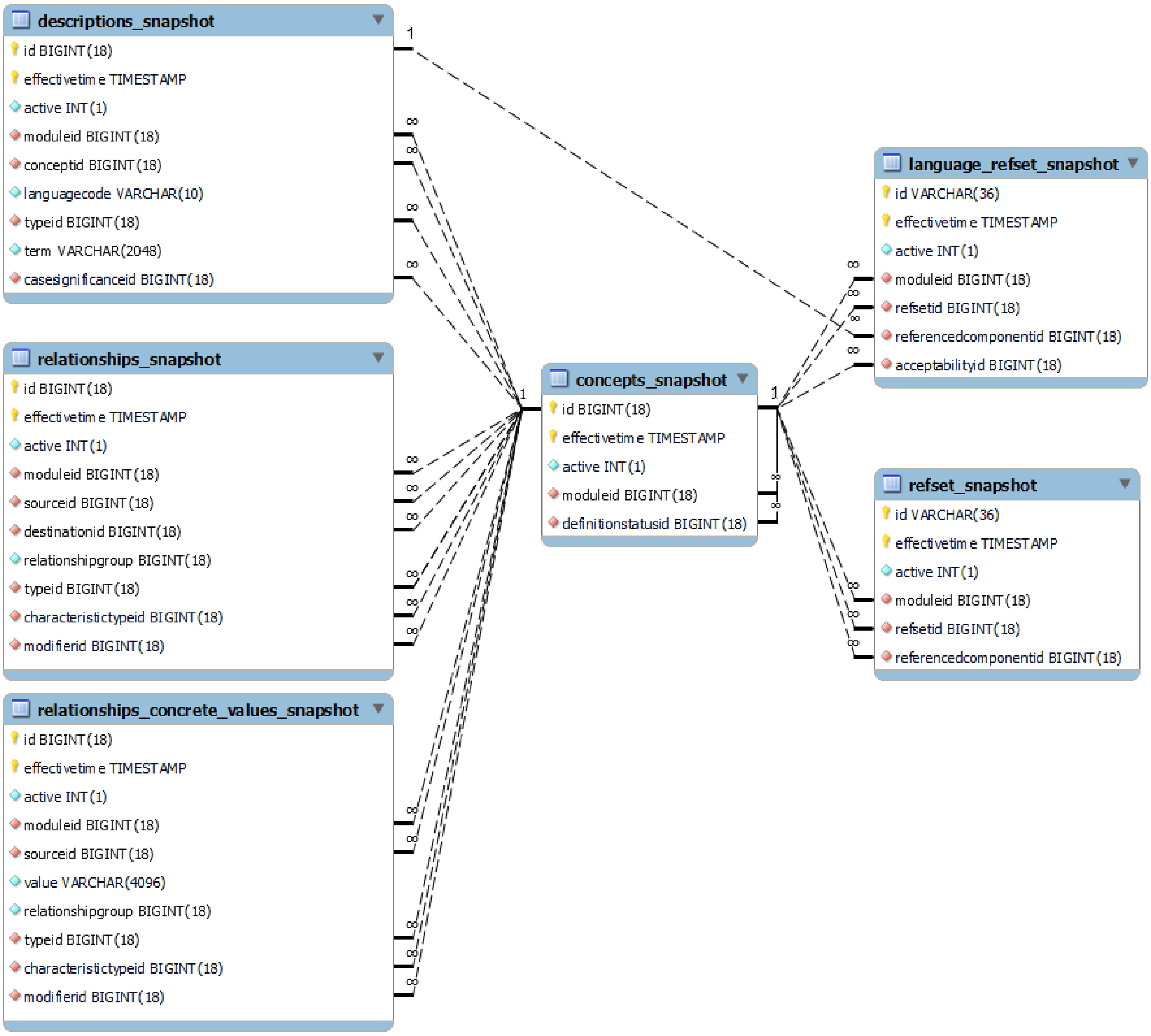


Figure 16: Entity-relationship diagram for sample schema

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[22] 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**

[22] SQL = Structured Query Language.

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 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.

- **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.

- **Reference set table naming convention**

  In the schema, tables containing reference sets are named according to their structure as outlined in the *SNOMED CT Release File Specifications* (SNOMED International, 2024). This allows for a compact and flexible database structure.

  However, these names are not intuitive when writing queries, so the following table lists the reference set table names within the sample schema with examples of what they contain (not all content has been listed).

Table 9: Sample schema table names with reference set content

Table name	Reference set name	Reference set ID
ccrefset_snapshot	*Dose route and form extended association reference set*	810501000168103
crefset_snapshot	*POSSIBLY EQUIVALENT TO association reference set*	900000000000523009
	REPLACED BY association reference set	900000000000526001
	SAME AS association reference set	900000000000527005
irefset_snapshot	*ARTG Id reference set*	11000168105

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
```

With a current working directory of <some-root-directory>, start a MySQL session.[23] 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 **sctau-sample-scripts-master/schema/2_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 2025:

```
<release-files>/Snapshot/Terminology/sct2_Concept_Snapshot_AU1000036_20250131.txt
```

Changes to

```
C:/Downloads/NCTS_SCT_RF2_DISTRIBUTION_32506021000036107-20240531-
ALL/Snapshot/Terminology/sct2_Concept_
Snapshot_AU1000036_20160131.txt
```

Once the SNOMED CT-AU release data has been successfully imported, the content can be retrieved using the sample queries contained within the **/AMTv4-sample-scripts/sql** directory.

[23] Instructions on how to open a mysql session and how to execute commands are available on the MySQL website.

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 **sctau-sample-scripts-master/sql/** directory.

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

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 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 conceptId = 248713000
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 rs.refsetid= 11000036103 -- ID of Adverse reaction type refset
AND d.typeid = 900000000000013009 -- Synonym
AND adrs.refsetid = 32570271000036106 -- ADRS
AND adrs.acceptabilityid=900000000000548007 -- ID of Preferred Term
AND c.active=1
AND d.active=1
AND rs.active=1
AND ADRS.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
refset_active.refsetid AS "Reference Set ID",
desc_active.term AS "Name of Reference Set",
refset_active.member_count AS "No of Members"
FROM
       (SELECT
       term,id,conceptid
       FROM descriptions_snapshot AS ds
       WHERE active=1 AND typeid = 900000000000013009 -- Synonym
       ) 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 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
JOIN (SELECT sourceId
             FROM transitive_closure
             WHERE destinationId=106112009 -- Fetal finding
             ) AS ffd
             ON c.id=ffd.sourceid
JOIN descriptions_snapshot AS d
          ON c.id=d.conceptid AND d.typeid = 900000000000013009 -- Synonym
JOIN language_refset_snapshot AS adrs
             ON d.id=adrs.referencedComponentId
AND adrs.refsetId = 32570271000036106  -- Australian dialect refset
WHERE 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 d.typeid = 900000000000013009 -- Synonym
AND adrs.refsetId = 32570271000036106  -- Australian dialect refset
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 d.typeid = 900000000000013009 -- Synonym
AND adrs.refsetId = 32570271000036106  -- Australian dialect refset
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

The AMT concept model is focused 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 **/AMTv4-sample-scripts/sql/AMT_use_cases.sql**.

9.5.1 Identifying members of notable reference sets ("notable concepts")

To obtain all notable concepts of a given type (e.g. *Medicinal product pack*) 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*|.

This can be queried as follows:

```
SELECT member.referencedcomponentid
FROM refset_snapshot AS member
WHERE member.refsetid = 929360081000036101 -- Medicinal product pack refset
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.

9.5.2 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 use 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 Pack Size Details

In order to extract the Pack Size Details (quantity and volume) for a given MPP, a query would need to

- Determine the pack size (quantity or volume) of MPUU in a given MPUU
- Navigate the MPP contains MPUU relationships
- Determine the pack size (if applicable) of any containered MPUUs within.
  (Maybe be recursive for combination packs).

By writing this query once to populate a derived table, the task of creating business queries to extract Pack Size Details is greatly simplified. An example of this query for an MPP is shown below:

Let’s first create a table where the MPUU contains MPUU pairs have already been calculated.

```
DROP TABLE IF EXISTS v4_MPPhasMPUU;
CREATE TABLE v4_MPPhasMPUU AS
SELECT
    MPPhasMPUU.sourceId as mppid,
    get_PT(MPPhasMPUU.sourceId) as mppterm,
    MPPhasMPUU.destinationId as mpuuid,
    get_PT(MPPhasMPUU.destinationId) as mpuuterm

FROM relationships_snapshot MPPhasMPUU
    WHERE MPPhasMPUU.active = 1
        AND MPPhasMPUU.typeid IN (774160008,999000081000168101) -- Contains
clinical drug / Contains device
    AND MPPhasMPUU.sourceId in (SELECT referencedComponentId FROM
refset_snapshot WHERE refsetId = 929360081000036101 AND active = 1) -- MPP
refset
    AND MPPhasMPUU.destinationId in (SELECT referencedComponentId FROM
refset_snapshot WHERE refsetId = 929360071000036103 AND active = 1); -- MPUU
refset
```

Now this table can use concreate_value_relationships and relationships to get the pack size (value and units, respectively), of the MPP and then the MPUU (if applicable, including container type))

```
DROP TABLE IF EXISTS v4_Packs_Size_Details;
CREATE TABLE v4_Packs_Size_Details AS
SELECT
    mppid,
    mppterm,
    mpuuid,
    mpuuterm,

    MPUUHasContainerType.destinationId as MPUUContainerTypeId,
    get_PT(MPUUHasContainerType.destinationId) as MPUUContainerTypeTerm,

    MPUUPackSizeValue.value as MPUUPackSizeValue,
    MPUUPackSizeUnit.destinationId as MPUUPackSizeUnitId,
    get_PT(MPUUPackSizeUnit.destinationId) as MPUUPackSizeUnitTerm,

    MPPPackSizeValue.value as MPUUinMPPQuantityValue,
    MPPPackSizeUnit.destinationId as MPUUinMPPQuantityunitid,
    get_PT(MPPPackSizeUnit.destinationId) as MPUUinMPPQuantityUnitterm

FROM v4_MPPhasMPUU MPPhasMPUU
LEFT JOIN relationships_snapshot MPUUHasContainerType
        on MPPhasMPUU.mpuuid = MPUUHasContainerType.sourceId
               AND MPUUHasContainerType.active = 1
        AND MPUUHasContainerType.typeId = 30465011000036106 -- Has container
type (attribute)

        LEFT JOIN relationships_concrete_values_snapshot MPUUPackSizeValue
        on MPPhasMPUU.mpuuid = MPUUPackSizeValue.sourceId
               AND MPUUPackSizeValue.typeid = 1142142004 -- Has pack size
(attribute)
        AND MPUUPackSizeValue.active = 1

    LEFT JOIN relationships_snapshot MPUUPackSizeUnit
        on MPUUPackSizeValue.sourceId = MPUUPackSizeUnit.sourceId
        AND MPUUPackSizeValue.relationshipgroup =
MPUUPackSizeUnit.relationshipgroup
               AND MPUUPackSizeUnit.typeid = 774163005 -- Has pack size unit
(attribute)
        AND MPUUPackSizeUnit.active = 1

    LEFT JOIN relationships_concrete_values_snapshot MPPPackSizeValue
        on MPPhasMPUU.mppid = MPPPackSizeValue.sourceId
               AND MPPPackSizeValue.typeid = 1142142004 -- Has pack size
(attribute)
        AND MPPPackSizeValue.active = 1

    LEFT JOIN relationships_snapshot MPPPackSizeUnit
        on MPPPackSizeValue.sourceId = MPPPackSizeUnit.sourceId
        AND MPPPackSizeValue.relationshipgroup =
MPPPackSizeUnit.relationshipgroup
               AND MPPPackSizeUnit.typeid = 774163005 -- Has pack size unit
(attribute)
        AND MPPPackSizeUnit.active = 1;
```

Although complex, subsequent queries can look up this data directly. Using the MPP 27145011000036106|Mesna 400 mg/4 mL injection, 15 x 4 mL ampoules| As an example:

```
SELECT * FROM v4_Packs_Size_Details
where mppid = 27145011000036106;
```

We get a single row describing:

- The MPP contains 15 units of “Mesna 400 mg/4 mL injection, ampoule”
- And that each of these MPUUs is a 4 mL vial.

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 contains MPUU relationship, the MPUU Has BoSS relationship, and two possible “strength” representations – “Has total quantity” and/or “Has concentration”. 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:

```
-- CREATE Table for v4_ingredient_strength
-- This table lists all the MPPs, their MPUUs AND the corresponding ingredients
(Active Ingredient AND BoSS) AND strengths - as either  "total quantity" in the
MPUU, "Concentration" or both.
DROP TABLE IF EXISTS v4_ingredient_strength;
CREATE TABLE v4_ingredient_strength AS
SELECT
        mppid,
    mppterm,
    mpuuid,
    mpuuterm,
    hasIngredient.destinationid as substanceid,
    get_PT(hasIngredient.destinationid) as substanceterm,
    hasBoSS.destinationid as bossid,
    get_PT(hasBoSS.destinationid) as bossterm,
    IF(TotalQuantityValue.value is null, '', TotalQuantityValue.value)
TotalQuantity,
    TotalQuantityUnit.destinationId as TotalQuantityUnitId,
    get_PT(TotalQuantityUnit.destinationid) as TotalQuantityUnitTerm,
    IF(ConcentrationValue.value is null, '', ConcentrationValue.value)
ConcentrationValue,
    ConcentrationUnit.destinationId as ConcentrationUnitId,
    get_PT(ConcentrationUnit.destinationid) as ConcentrationUnitTerm

FROM v4_MPPhasMPUU MPPhasMPUU

    LEFT JOIN relationships_snapshot hasIngredient
        on MPPhasMPUU.mpuuid = hasIngredient.sourceId
        AND hasIngredient.typeId in (127489000,762949000) -- Has active
ingredient (attribute) / Has precise active ingredient (attribute)
        AND hasIngredient.active = 1

    LEFT JOIN relationships_snapshot hasBoSS
        on hasIngredient.sourceId = hasBoSS.sourceId
        AND hasIngredient.relationshipgroup = hasBoSS.relationshipgroup
        AND hasBoSS.typeId = 732943007 -- Has basis of strength substance
(attribute)
        AND hasBoSS.active = 1
    LEFT JOIN relationships_concrete_values_snapshot TotalQuantityValue
        on hasBoSS.sourceId = TotalQuantityValue.sourceId
        AND hasBoSS.relationshipgroup = TotalQuantityValue.relationshipgroup
               AND TotalQuantityValue.typeid = 999000041000168106 -- Has total
quantity value (attribute)
        AND TotalQuantityValue.active = 1

    LEFT JOIN relationships_snapshot TotalQuantityUnit
        on hasBoSS.sourceId = TotalQuantityUnit.sourceId
        AND hasBoSS.relationshipgroup = TotalQuantityUnit.relationshipgroup
               AND TotalQuantityUnit.typeid = 999000051000168108 -- Has total
quantity unit (attribute)
        AND TotalQuantityUnit.active = 1


    LEFT JOIN relationships_concrete_values_snapshot ConcentrationValue
        on hasBoSS.sourceId = ConcentrationValue.sourceId
        AND hasBoSS.relationshipgroup = ConcentrationValue.relationshipgroup
               AND ConcentrationValue.typeid = 999000021000168100 -- Has
concentration strength value (attribute)
        AND ConcentrationValue.active = 1

    LEFT JOIN relationships_snapshot ConcentrationUnit
        on hasBoSS.sourceId = ConcentrationUnit.sourceId
        AND hasBoSS.relationshipgroup = ConcentrationUnit.relationshipgroup
               AND ConcentrationUnit.typeid = 999000031000168102 -- Has
concentration strength unit (attribute)
        AND ConcentrationUnit.active = 1;
```

The determine the strength associated with an MPP is now queried as:

```
SELECT *
FROM v4_ingredient_strength
WHERE mppid = 27145011000036106;
```

This query returns a row for each ingredient in associated MPP:

- Mesna 400 mg (total quantity, in MPUU)
- Mesna 100 mg/mL (concentration)

9.5.2.3 Query for prescribing use case

The goal of the Prescribing use case in the *AMT Concept Model and Business Use Cases* (Australian Digital Health Agency, 2017) 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 amoxicillin, 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 **v4_mpp_to_tpp** and **v4_ingredient_strength** as follows:

```
SELECT
       v4_mpp_to_tpp.mppid,
       v4_mpp_to_tpp.mppterm,
       v4_mpp_to_tpp.tppid,
       v4_mpp_to_tpp.tppterm
FROM v4_mpp_to_tpp

JOIN v4_ingredient_strength
       ON v4_mpp_to_tpp.mppid = v4_ingredient_strength.mppid

WHERE v4_ingredient_strength.substanceterm REGEXP (@search_term:='(^|[^a-zA-
Z]+)amox')

OR v4_mpp_to_tpp.mppterm REGEXP @search_term
OR v4_mpp_to_tpp.tppterm LIKE @search_term
;
```

The above query primarily searches the **v4_ingredient_strength** table as it is already populated with terms for the MPPs, MPUUs and substances within them. We additionally join to the **v4_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. 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.

9.5.2.4 Query for dispensing use case

The goal of the Dispensing use case in the *AMT Concept Model and Business Use Cases* (Australian Digital Health Agency, 2017) 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 hard capsule, 20|*, which is a TPP concept with the Preferred Term “Amoxil 250 mg hard capsule, 20 capsules”. The prescription indicates that a generic alternative is acceptable.

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).[24] Using the derived model table **v4_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,
    get_PT(ctpps.sourceid) AS substitutectppterm
FROM v4_mpp_to_tpp AS tpp1

JOIN v4_mpp_to_tpp AS substitutetpp
     ON tpp1.mppid = substitutetpp.mppid
AND tpp1.tppid != substitutetpp.tppid
AND tpp1.tppid = 12809011000036105 -- Amoxil 250 mg hard capsule, 20

JOIN relationships_snapshot AS ctpps
     ON substitutetpp.tppid = ctpps.destinationid
AND ctpps.sourceId in (SELECT referencedComponentId FROM refset_snapshot where
refsetId = 929360051000036108)
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.

9.5.2.5 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.

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

```
SELECT
    v4_ingredient_strength.mppid,
    v4_ingredient_strength.mppterm,
    v4_ingredient_strength.bossterm,
    get_PT(hasDoseForm.destinationid)
FROM v4_ingredient_strength

JOIN relationships_snapshot AS hasDoseForm
     ON hasDoseForm.sourceid = v4_ingredient_strength.mpuuid
     AND hasDoseForm.typeid = 411116001 -- Has manufactured dose form
(attribute)
     AND hasDoseForm.active = 1

WHERE v4_ingredient_strength.mppid in (
    26624011000036107,     -- 'amoxicillin 100 mg/mL oral...'
    751341000168105,       -- 'Hydrocortisone 1% + clotri...'
    834551000168104,       -- 'Aspirin 325 mg + codeine p...'
    28049011000036105      -- 'Peginterferon alfa-2a 135 ...'
)
;
```

Table 10: Results from a sample query that extracts MPP concepts and their associated dose forms

MPP ID	MPP PT	Substance	Dose Form
751341000168105	Hydrocortisone 1% + clotrimazole 1% cream, 15 g	Hydrocortisone	Cream
751341000168105	Hydrocortisone 1% + clotrimazole 1% cream, 15 g	Clotrimazole	Cream
834551000168104	Aspirin 325 mg + codeine phosphate hemihydrate 30 mg tablet, 20	Codeine phosphate hemihydrate	Tablet
834551000168104	Aspirin 325 mg + codeine phosphate hemihydrate 30 mg tablet, 20	Aspirin	Tablet
26624011000036107	Amoxicillin 100 mg/mL powder for oral liquid, 20 mL	Amoxicillin	Powder for oral liquid
28049011000036105	Peginterferon alfa-2a 135 microgram/0.5 mL injection, 4 x 0.5 mL syringes	Peginterferon alfa-2a	Injection

9.5.3 Schedule 8 medicines

The *Schedule 8 medications reference set* is a simple type reference set, to support the Electronic Recording and Reporting of Controlled Drugs (ERRCD) initiative. The ERRCD aims to develop a nationally consistent system to collect and report data relating to the prescribing and dispensing of all Schedule 8 (S8) Controlled Drugs to complement and support the current regulatory controls required by states and territories.

This reference set contains CTPP and TPUU concepts for S8 drugs, based on the Schedule information held in the ARTG. The following query can help to identify which of the dispensed drugs (if any) are listed as S8.

```
SELECT
    referencedcomponentid

FROM refset_snapshot

WHERE referencedcomponentid IN (
    779911000168108, -- Artige 10 mg uncoated tablet, 100, blister pack
    933214301000036108 -- Tralen 50 mg film-coated tablet, 14, blister pack
    )
AND refsetid = 1050951000168102 -- Schedule 8 medications reference set
AND active = 1
```

Further queries are provided in the sample script bundle that append ARTG ID to the CTPPs in the reference set, by using the *ARTG ID reference set*, and derive TPPs and MPUUs if required for product matching or reporting purposes.

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[25] 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 “Testing and traversing subtype relationships” of the *SNOMED CT Terminology Services Guide* (SNOMED International) 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 7.5.2 “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.

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

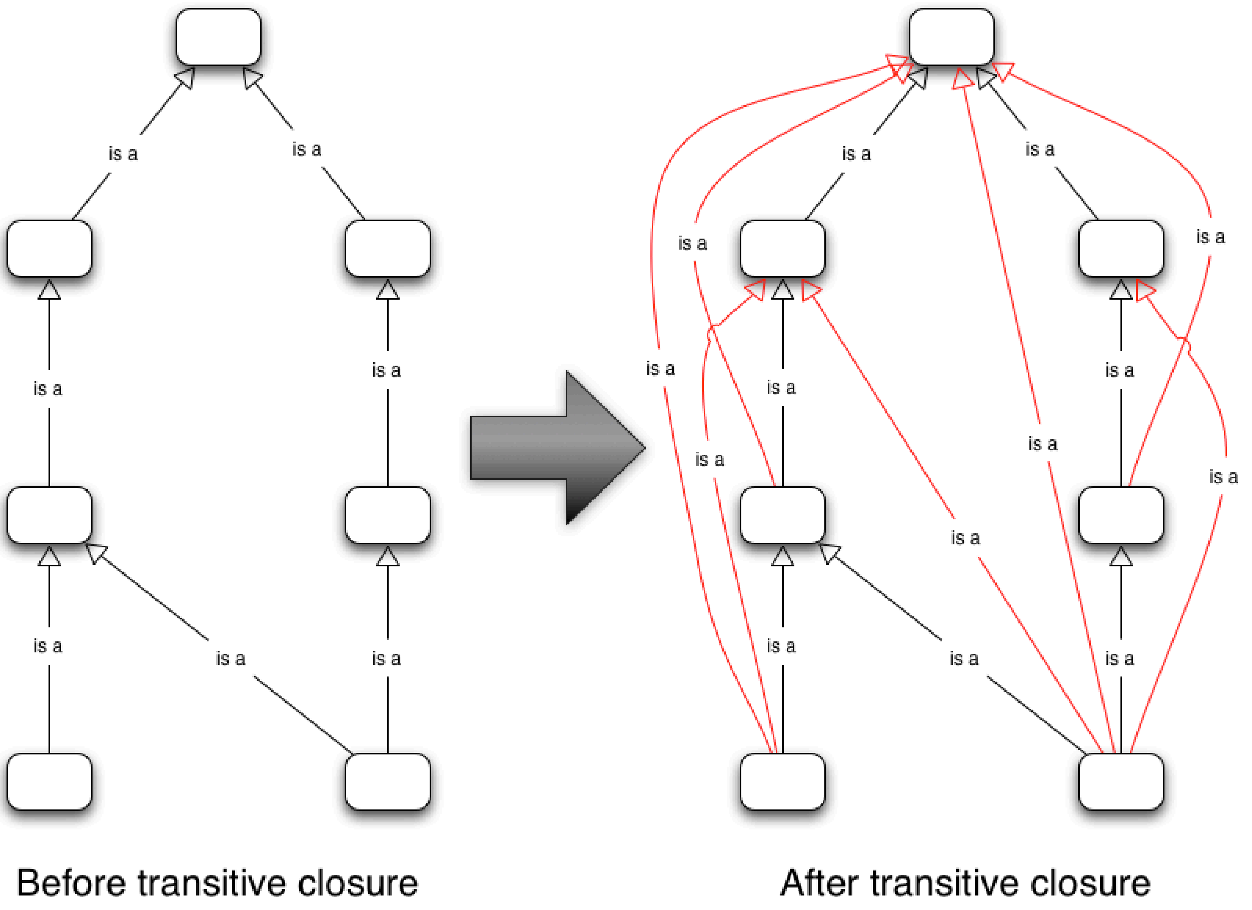


Figure 17: Example of a transitive closure

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 11: 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 16 IS A relationships[26] 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 12: 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	*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*

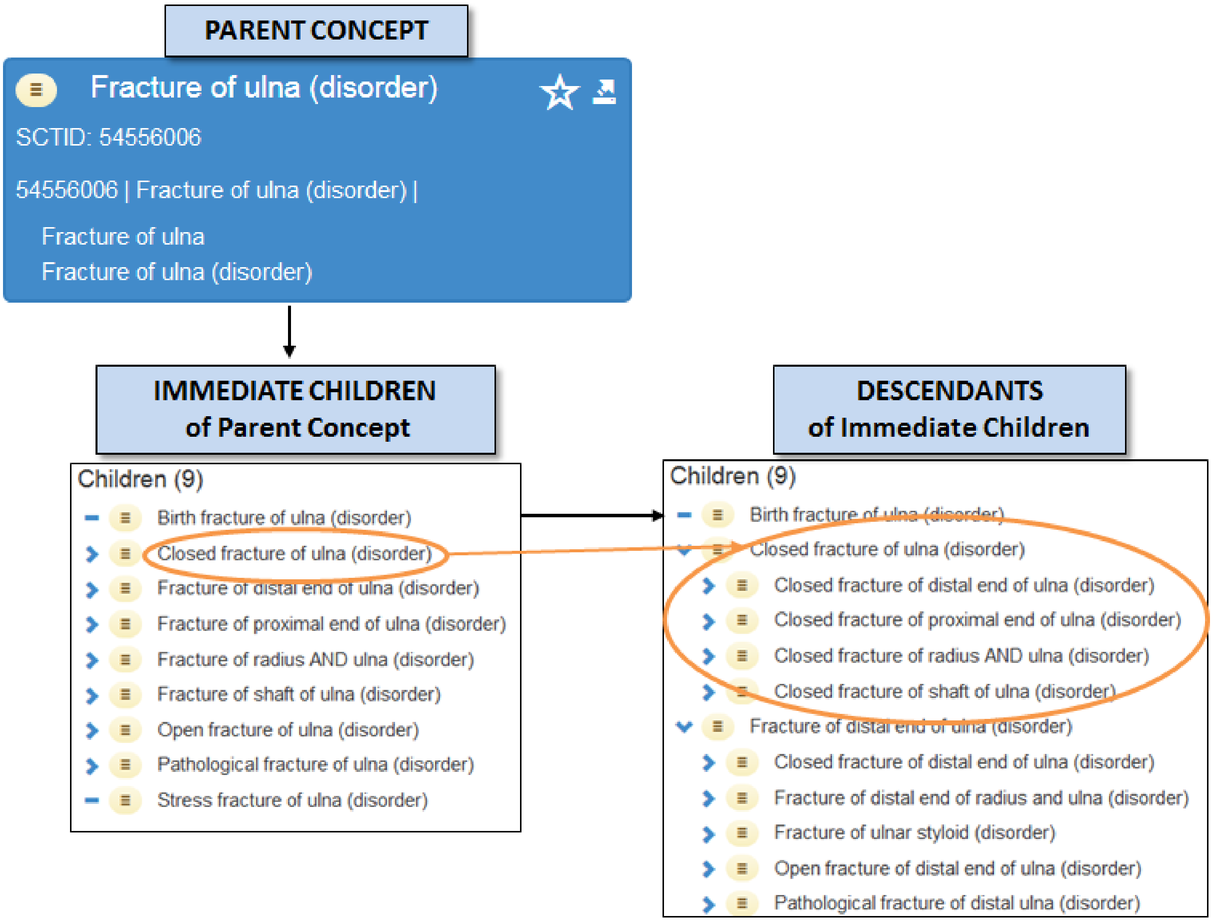
[26] Based on SNOMED CT AU May 2017 data. Different releases may produce different results.

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.[27] 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 18: Descendants of "Fracture of ulna"

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*|.

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

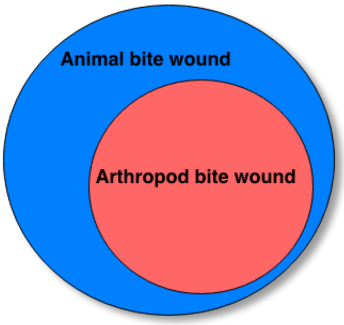


Figure 19: Relationship between Animal bite wound and Arthropod bite wound concepts

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 sub-hierarchies

Concepts in SNOMED CT-AU are organised in a polyhierarchy[28], 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.

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

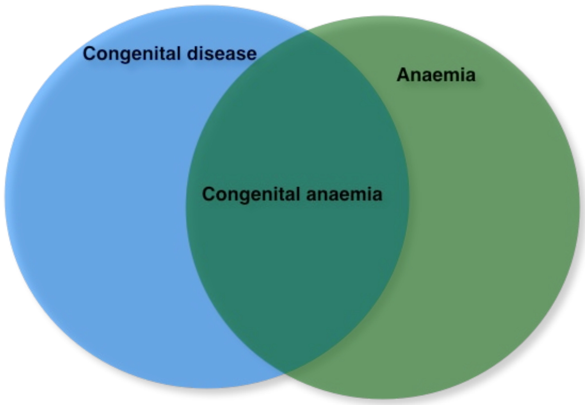


Figure 20: Intersection of Congenital diseases and Anaemias

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; or to
- map to or align with other code systems and classifications.

Regardless of implementation type or technologies, terminology content will have to be regularly updated to a new version of SNOMED CT-AU and the AMT within 30 or 90 days of a release, depending on your use case. See Clause 6 of the Australian National Terminology Licence[29] for details.

Requests for further terminology and support queries should be directed to help@digitalhealth.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.

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

- Providing tools for viewing SNOMED CTAU and its content, such as reference sets.
- Offering expert skills and education to SNOMED CTAU 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 online NCTS Request Submission forms.

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.

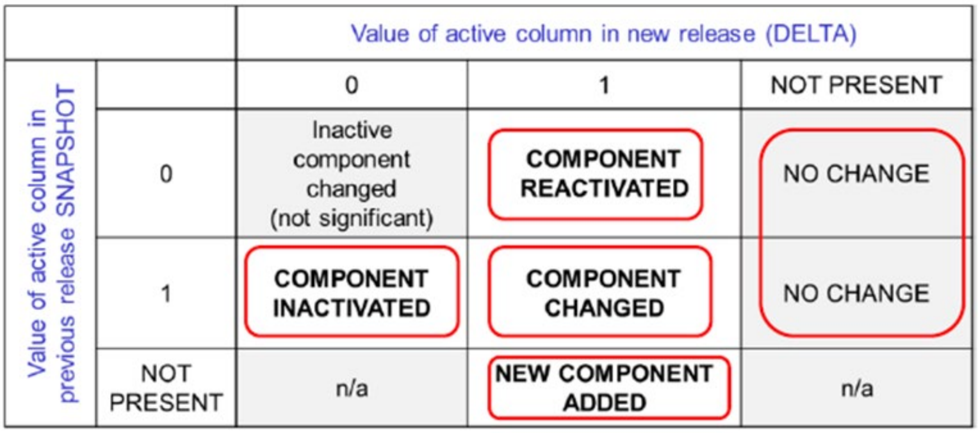
[29] Available from the NCTS Document Library.

11.2 Identifying changes

In order to identify content changes that have occurred for between two specific releases, a Delta release type can be used.

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

- "previous release SNAPSHOT" refers to the "older" release used generate a Delta.
- "new release DELTA" refers to the delate generated of the "newer" release.



		Value of active column in new release (DELTA)		
Value of active column in previous release SNAPSHOT		0	1	NOT PRESENT
	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 21: 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 13: 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.
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 7.5.2 “Transitive Closure Implementation” in the *SNOMED CT Terminology Services Guide* (SNOMED International).

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:

https://confluence.ihtsdotools.org/display/DOCTSG/7.5.2+Transitive+closure+imple
mentation
---------------------------------------------------------------------*/

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 subset of the content within SNOMED CT-AU. 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 dependency 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 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.*[30]

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 content that comprises the SNOMED CT-AU is composed of two modules for the Australian extension and Third Party reference sets, as well as the core edition modules.

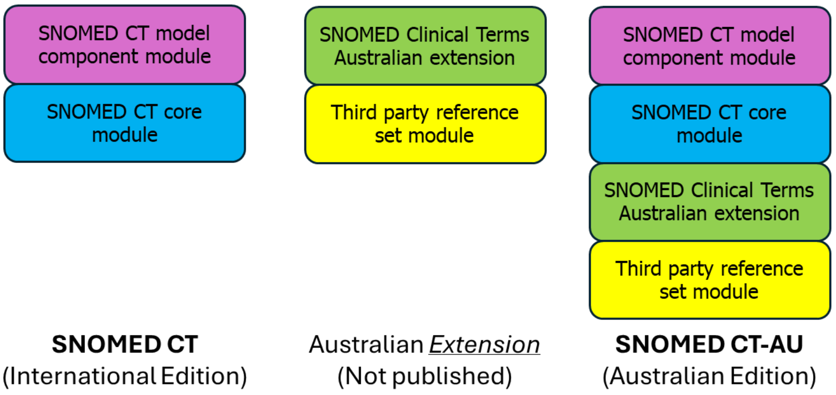

SNOMED CT model component module
SNOMED CT core module
SNOMED Clinical Terms Australian extension
Third party reference set module
SNOMED CT model component module
SNOMED CT core module
SNOMED Clinical Terms Australian extension
Third party reference set module
SNOMED CT
(International Edition)
Australian Extension
(Not published)
SNOMED CT-AU
(Australian Edition)

Figure 22: Release modules in terminology releases

The *Module dependency reference set*[31] 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.

[30] *SNOMED CT Glossary* (SNOMED International).

[31] Section 4.2.11 "Module Dependency Reference Set" of the *SNOMED CT Release File Specification* (SNOMED International, 2024).

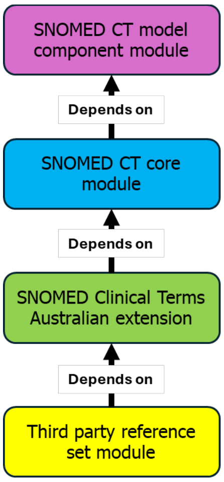

SNOMED CT model component module
Depends on
SNOMED CT core module
Depends on
SNOMED Clinical Terms Australian extension
Depends on
Third party reference set module

Figure 23: 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. A module cannot be used or implemented without all other modules it depends on.

(Historical) AMT Module

With the introduction of AMT v4 model. AMT content developed a strong binding to SNOMED CT-core content – specifically through referencing of core concepts like substances, units of measures and dose forms, in defining relationships. As a result, AMT can no longer be used without SNOMED CT core – and the Australian extension was consolidated into a single module.

Appendix C Metadata cheat sheet

C.1 Description types

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

C.2 ADRS preferences

Name of description type	Acceptability
Preferred	900000000000548007
Acceptable	900000000000549004

C.3 Relationship types

Name of relationship	Relationship identifier
Access	260507000
Associated finding	246090004
Associated morphology	116676008
Associated procedure	363589002
Associated with	47429007
Before	288556008
Causative agent	246075003
Characterizes	704321009
Clinical course	263502005
Component	246093002
Contains clinical drug	774160008

Name of relationship	Relationship identifier
Contains device	999000081000168101
Contains packaged clinical drug	999000011000168107
Developed by	1445121000168107
Direct device	363699004
Direct morphology	363700003
Direct site	704327008
Direct substance	363701004
Due to	42752001
During	371881003
Finding context	408729009
Finding informer	419066007
Finding method	418775008
Finding site	363698007
Following	255234002
Has absorbability	1148969005
Has active ingredient	127489000
Has basic dose form	736476002
Has BoSS	732943007
Has coating material	1148967007

Name of relationship	Relationship identifier
Has compositional material	840560000
Has concentration strength denominator unit	733722007
Has concentration strength numerator unit	733725009
Has concentration strength unit	999000031000168102
Has container type	30465011000036106
Has denominator units	700000071000036103
Has device intended site	836358009
Has device type	999000061000168105
Has disposition	726542003
Has dose form administration method	736472000
Has dose form intended site	736474004
Has dose form release characteristic	736475003
Has dose form transformation	736473005
Has filling	827081001
Has focus	363702006

Name of relationship	Relationship identifier
Has ingredient qualitative strength	1149366004
Has intent	363703001
Has interpretation	363713009
Has licence	1445111000168100
Has manufactured dose form	411116001
Has numerator units	700000091000036104
Has pack size unit	774163005
Has precise active ingredient	762949000
Has presentation strength denominator unit	732947008
Has presentation strength numerator unit	732945000
Has product name	774158006
Has realisation	719722006
has Route of administration	410675002
Has specimen	116686009
Has state of matter	736518005
Has surface texture	1148968002

Name of relationship	Relationship identifier
Has target population	1149367008
Has total quantity unit	999000051000168108
Has unit of presentation	763032000
Indirect device	363710007
Indirect morphology	363709002
Inherent location	718497002
Inheres in	704319004
Interprets	363714003
Is a	116680003
Is modification of	738774007
Is sterile	1148965004
Laterality	272741003
Measurement method	370129005
Method	260686004
Occurrence	246454002
Pathological process	370135005
Plays role	766939001
Precondition	704326004
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
Process acts on	1003735000
Process duration	704323007
Process extends to	1003703000
Process output	704324001
Property	370130000
Recipient category	370131001
Relative to	704325000
Relative to part of	719715003
Revision status	246513007
Scale type	370132008
Severity	246112005
Specimen procedure	118171006
Specimen source identity	118170007
Specimen source morphology	118168003
Specimen source topography	118169006

Name of relationship	Relationship identifier
Specimen substance	370133003
Subject relationship context	408732007
Surgical approach	424876005
Technique	246501002
Temporal context	408731000
Temporally related to	726633004
Time aspect	370134009
Towards	704320005
Units	246514001
Using access device	425391005
Using device	424226004
Using energy	424244007
Using substance	424361007

C.4 Concrete value relationship types

Name of relationship	Relationship identifier
Count of active ingredient	1142140007
Count of clinical drug type	1142143009

Name of relationship	Relationship identifier
Count of contained component ingredient	999000131000168101
Count of contained package types	999000091000168103
Count of device type	999000101000168108
Has concentration strength value	999000021000168100
Has other identifying information	999000001000168109
Has pack size value	1142142004
Has total quantity value	999000041000168106

Appendix D Description logic and reasoning with SNOMED CT

D.1 Defined and Primitive concepts

In SNOMED CT concepts may be considered “defined” or “primitive”, which indicates if the concept’s set of relationships[32] are sufficient to define the concept. This is SNOMED CT’s mechanism for representing necessity and sufficiency.

Defined concept:	The concept’s set of relationships are necessary and sufficient to define the concept in terms of the other concepts in the terminology (that is, none of its defining characteristics are missing, and its set of relationships are unique).
Primitive concept:	While necessary, the concept’s set of relationships is not sufficient to distinguish the concept in terms of the other concepts in the terminology.

An example of a defined concept in the AMT is 90332006 | Product containing paracetamol (medicinal product) |. This concept’s definition consists of:

- an attribute |*Has active ingredient*| = |*Paracetamol (substance)*| to define the ingredient it contains; and
- an |*Is a*| relationship to the concept |*Medicinal product*| to specifying that it is a product.

This definition means that any concept that contains paracetamol and is also a type of medicinal product is therefore a subtype of the concept | Product containing paracetamol (medicinal product)|.

However, the concept |*Paracetamol (substance)*| is declared as primitive as its definition only has one relationship stating it is a subtype of |*Australian substance*|.

To define |*paracetamol (substance)*| within the AMT would require adding relationships and concepts defining the molecular structure of paracetamol, which are not relevant to the SNOMED CT-AU use cases. Therefore these relationships are omitted and *Substance* concepts in the AMT are declared primitive.

D.2 Translation to common knowledge representation languages

Most implementations will not need to reason with the content of the AMT or SNOMED CT; however it is possible if required.

The set of concepts, relationships and primitive/defined status can be translated into a common knowledge representation language for this purpose. SNOMED International provides scripts to translate between SNOMED CT release formats and common knowledge representation formats such as OWL (Web Ontology Language) and KRSS (Knowledge Representation System Specification).

[32] That is, those relationships where the concept is the source of the relationship.

Implementers who need to generate such knowledge representations are advised to contact the NCTS for assistance.

D.3 Open versus closed world assumption

Unlike most information models and databases, which operate under a closed world assumption, SNOMED CT terminologies operate under an open world assumption. Under a closed world assumption, any statement not known to be true is implicitly false. However, under an open world assumption, the omission of a statement simply means that it is unknown – and could be true or false. For example:

Statement:	Mary is a citizen of France.
Question:	Is Paul a citizen of France?
“Closed world” (for example, SQL) answer:	No.
“Open world” answer:	Unknown.

SNOMED CT and AMT concepts simply represent what is known, and do not include or preclude other conditions that may be true or false.

For example in SNOMED CT |*fracture of tibia*| obviously does not mean a fracture of the fibula or femur can be assumed. However, the omission of a statement about the fibula and femur does not rule out a fracture of fibula or femur either. The statement |*fracture of tibia*| simply states that a tibia fracture exists, and any state of the fibula, femur or any other bone in the body is unknown.

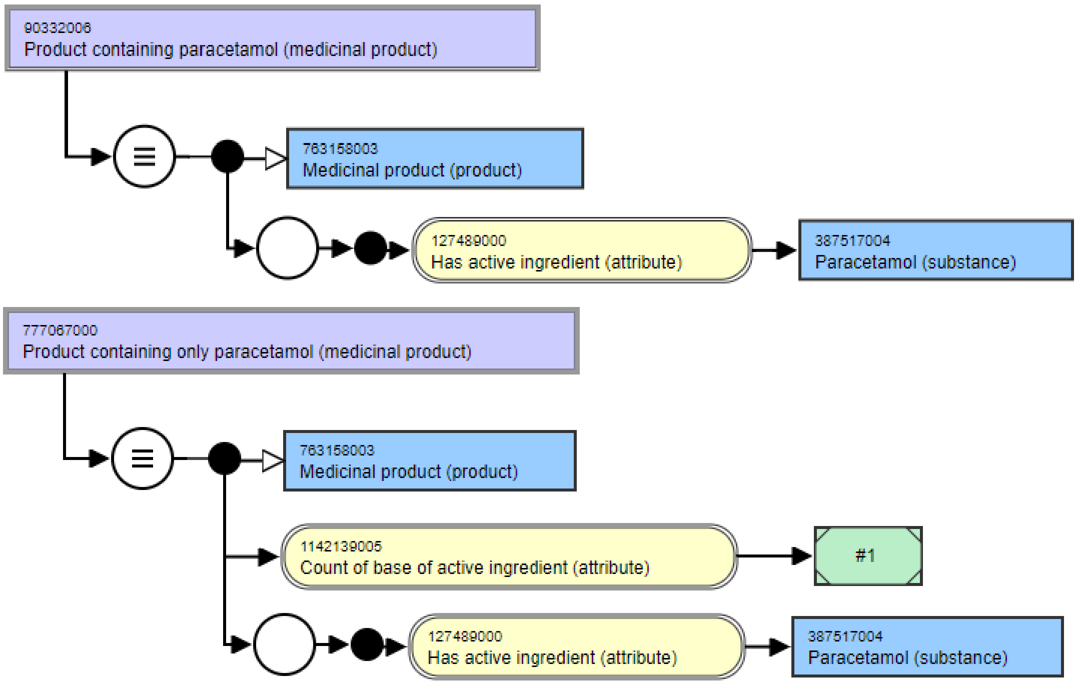
Similarly, for the concept | Product containing paracetamol (medicinal product)| has a relationship HAS INTENDED ACTIVE INGREDIENT to |*Paracetamol (substance)*|. This states the presence of at least paracetamol in the product; however it makes no statement about the presence or absence of any other ingredients (active or otherwise). Therefore | Codeine- and paracetamol-containing product| is a subtype of | Product containing paracetamol (medicinal product)| because at least paracetamol is present.

This differs from a closed world model which would assume | Product containing paracetamol (medicinal product)| contained only paracetamol and no other ingredients because no others were stated.

Existential and universal restrictions

Put simply, existential restrictions mean “at least” or “some”, whereas universal restrictions mean “for all” or “only”. SNOMED CT relationships are expressed using existential restrictions. Universal restrictions are currently not supported in SNOMED CT’s subset of description logic features.

To achieve the closed world semantics in within SNOMED CT, this (closed world) definition is simulated by specifying the counts of these specific properties. To illustrate, again consider the concept | *Product containing only paracetamol (medicinal product)* | for which only concepts containing paracetamol and no other active ingredients are considered to be subtypes. The definitions of these two concepts are shown below. **Note:** “ONLY” concepts are not considered part of the AMT 7 notable concepts and exist as part of the core international drug model.


90332006
Product containing paracetamol (medicinal product)
763158003
Medicinal product (product)
127489000
Has active ingredient (attribute)
387517004
Paracetamol (substance)
777067000
Product containing only paracetamol (medicinal product)
763158003
Medicinal product (product)
1142139005
Count of base of active ingredient (attribute)
#1
127489000
Has active ingredient (attribute)
387517004
Paracetamol (substance)

Appendix E Subpacks and combination packs

A product pack (MPP or TPP) always contains components (MPUUs or TPUUs) in a primary container. The primary container is the lowest-level container (non-ingestible) that immediately surrounds the medicinal product. Examples of a primary container are: *blister pack*, *bottle*, *vial* and *cartridge*.

Some products may also have a secondary container that envelops the components contained within one or more primary containers. An example of a secondary container is “carton”. The AMT does not include specific information on secondary containers but uses the container type “pack” for all CTPP representing secondary containers.

The component(s) within a secondary container may:

- have the same active ingredients, strength and form;
- have the same active ingredients, different strengths but similar form; or
- have different active ingredients, different strengths but similar form.

In AMT v4 the subpacks contained within a pack containing subpacks is specified in the model using either of the “*Contains packaged product*” relationships:

- 999000011000168107| Contains packaged clinical drug |
- 999000111000168106| Contains packaged device |

Each of these relationships is also grouped with a “Has pack size value” that specifies the quantity, and of course the actual product.

Note: In AMTv4 Subpacks and Combination packs use identical modelling. The following descriptions are provided to help with legacy understanding.

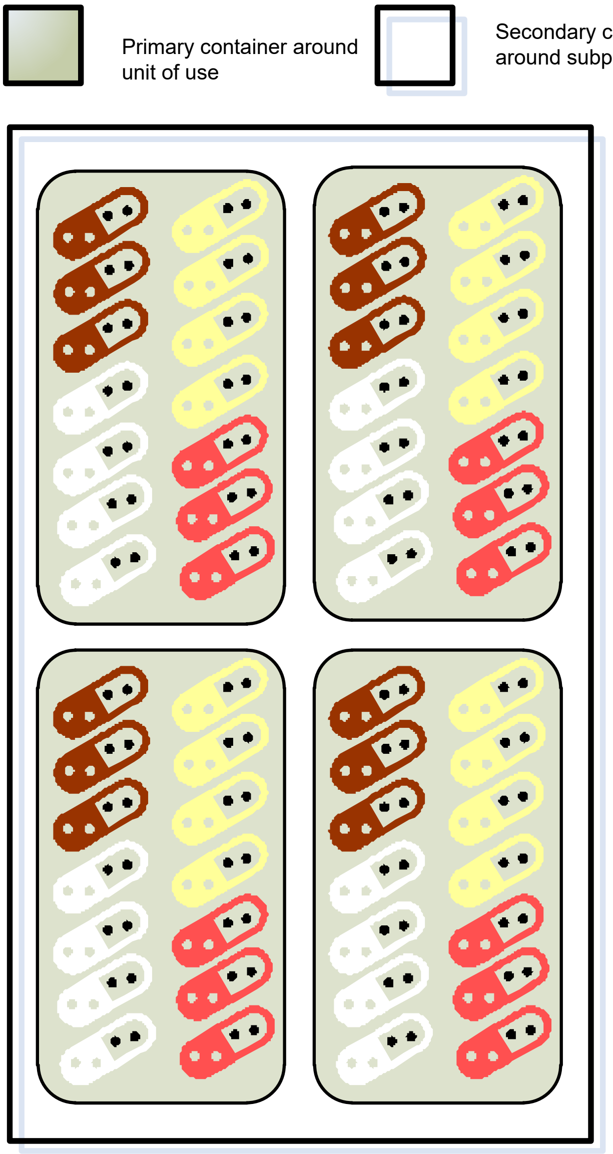
E.1 Subpacks

When there are multiple identical representations of the same component(s) within the same type of primary container, the product pack is said to have subpacks.

Subpacks are only represented for specific product categories in the AMT. The categories currently include some oral contraceptives, hormone replacement therapy products and any other multi-pack products that are presented in multiple subpacks. Only the primary container immediately enveloping the MPUUs is specified for these products (for example, blister pack). The secondary container is simply specified as “Pack”.

The following figure depicts an example of a pack containing four identical subpacks.

Figure 24: Abstract example of a product with 4 subpacks

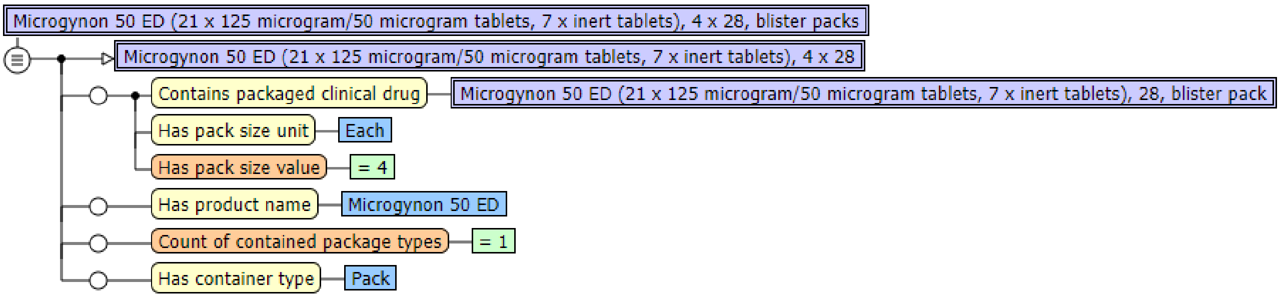

Primary container around unit of use
Secondary container around subpacks

Subpacks will be added when they are deemed to be required:

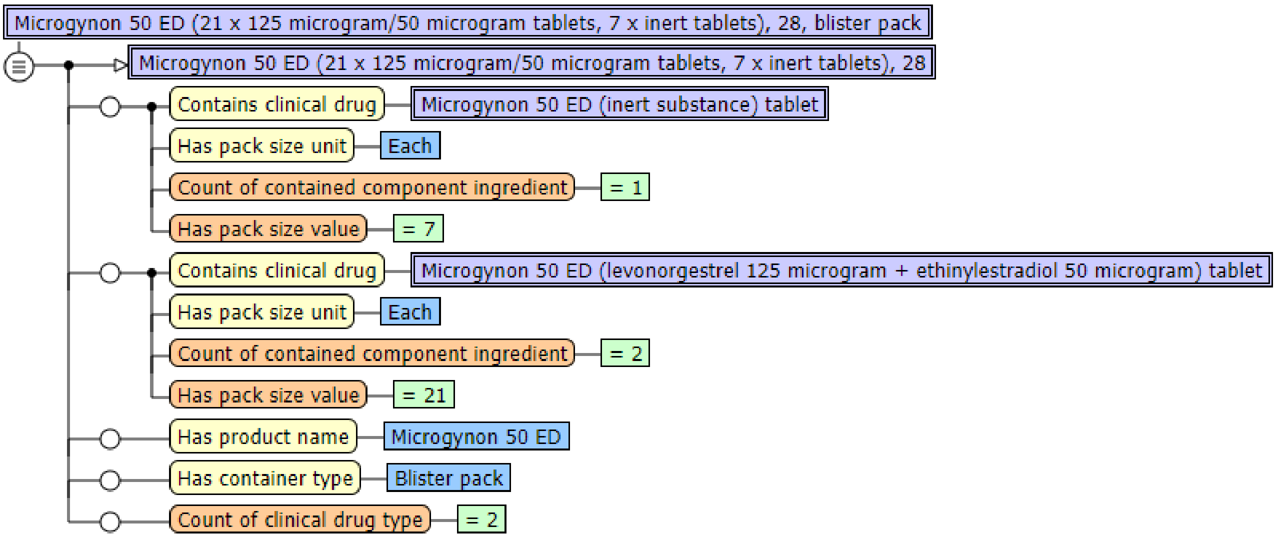
- for consistency;
- for clinical reasons; or
- when they are represented as subpacks in the Pharmaceutical Benefits Scheme (PBS).

The following is an example of a CTPP pack containing subpacks and its subpack CTPP respectively:

- *|Microgynon 50 ED (21 x 125 microgram/50 microgram tablets, 7 x inert tablets), 4 x 28, blister packs|*


Microgynon 50 ED (21 x 125 microgram/50 microgram tablets, 7 x inert tablets), 4 x 28, blister packs
Microgynon 50 ED (21 x 125 microgram/50 microgram tablets, 7 x inert tablets), 4 x 28
Contains packaged clinical drug
Microgynon 50 ED (21 x 125 microgram/50 microgram tablets, 7 x inert tablets), 28, blister pack
Has pack size unit
Each
Has pack size value
= 4
Has product name
Microgynon 50 ED
Count of contained package types
= 1
Has container type
Pack

- |*Microgynon 50 ED (21 x 125 microgram/50 microgram tablets, 7 x inert tablets), 28, blister pack*|.

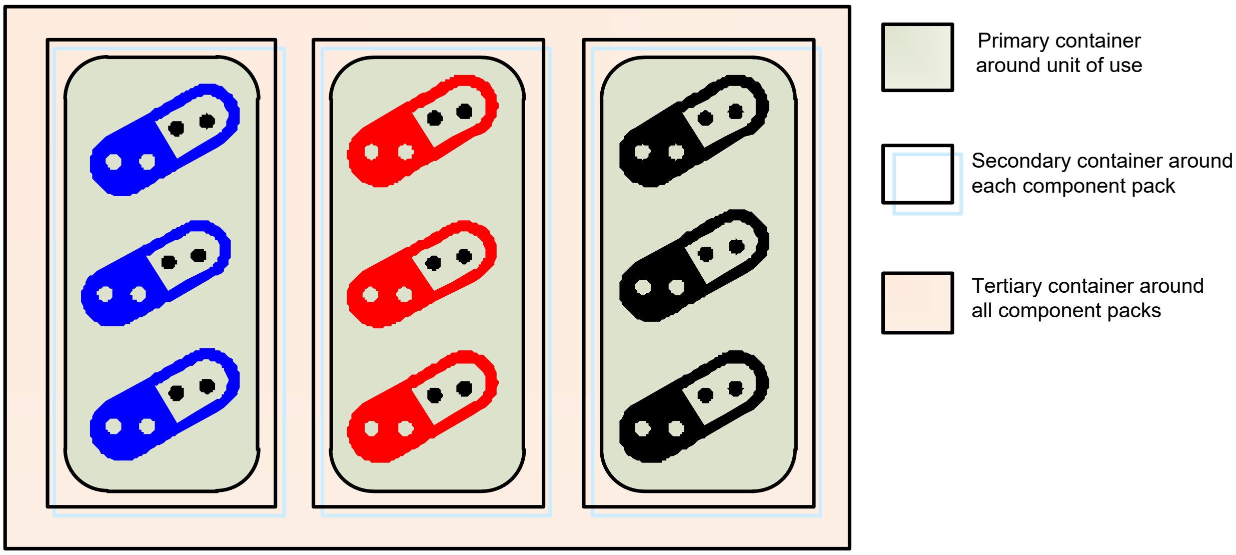

Microgynon 50 ED (21 x 125 microgram/50 microgram tablets, 7 x inert tablets), 28, blister pack
Microgynon 50 ED (21 x 125 microgram/50 microgram tablets, 7 x inert tablets), 28
Contains clinical drug
Microgynon 50 ED (inert substance) tablet
Has pack size unit
Each
Count of contained component ingredient
= 1
Has pack size value
= 7
Contains clinical drug
Microgynon 50 ED (levonorgestrel 125 microgram + ethinylestradiol 50 microgram) tablet
Has pack size unit
Each
Count of contained component ingredient
= 2
Has pack size value
= 21
Has product name
Microgynon 50 ED
Has container type
Blister pack
Count of clinical drug type
= 2

E.2 Combination packs

If a product pack contains multiple units of use, with each Unit of Use contained in a separate primary container, then the product pack is deemed to have component packs. The component packs within a combination product pack typically have different active ingredients. They may have the same form or have different forms.

The CTPP representing the combination product pack (that is, containing all the component packs) will have an associated generic container type of “pack”. The CTPP representing a component pack will have an associated specific container type such as *blister pack*, *bottle*, and *ampoule*. Only the primary container immediately enveloping the Unit of Use is modelled for these products (for example, blister pack). Secondary and tertiary containers are not modelled.

Figure 25: Abstract example of product with three component packs.

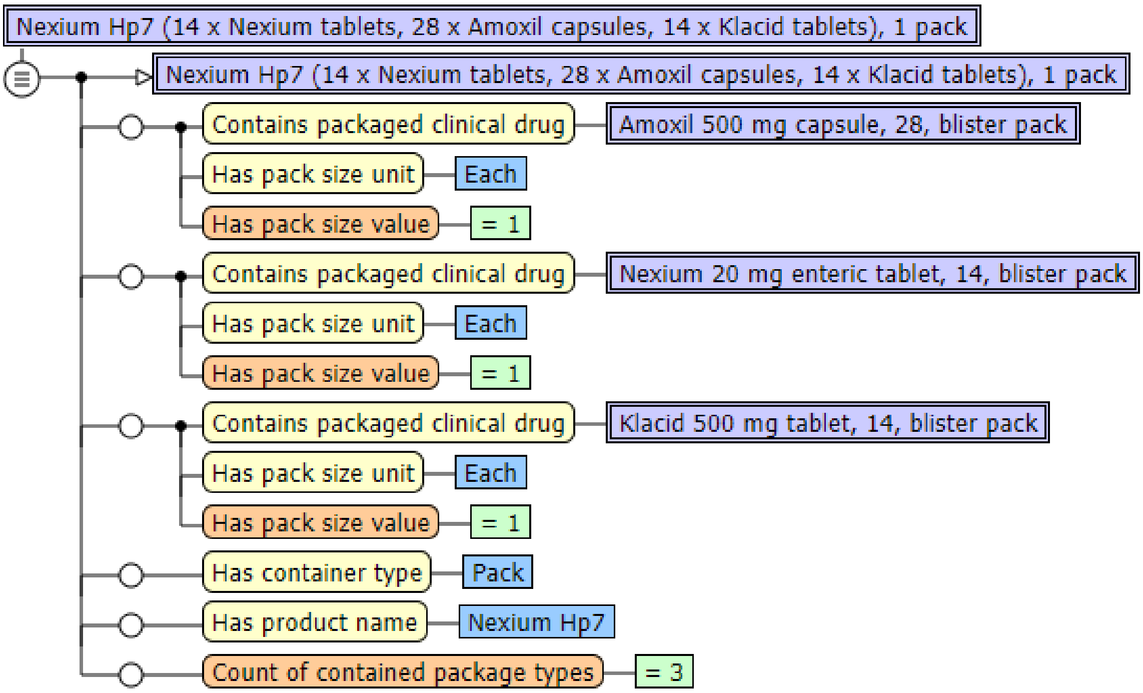

Primary container around unit of use
Secondary container around each component pack
Tertiary container around all component packs

For example, Nexium Hp7 is a combination pack and a multi-component product. It will have four associated CTPPs: one to represent each of the three component packs and an additional CTPP to represent the overall combination pack product.

Nexium Hp7's component pack CTPPs are as follows:

- |*Amoxil 500 mg hard capsule, 28, blister pack*|
- |*Klacid 500 mg film-coated tablet, 14, blister pack*|
- |*Nexium 20 mg enteric tablet, 14, blister pack*|

And its combination pack CTPP is | Nexium Hp7 (14 x Nexium tablets, 28 x Amoxil capsules, 14 x Klacid tablets), 1 pack |. That is, this is the overall product, the "combination pack" CTPP.



Nexium Hp7 (14 x Nexium tablets, 28 x Amoxil capsules, 14 x Klacid tablets), 1 pack

Nexium Hp7 (14 x Nexium tablets, 28 x Amoxil capsules, 14 x Klacid tablets), 1 pack

Contains packaged clinical drug — Amoxil 500 mg capsule, 28, blister pack
Has pack size unit — Each
Has pack size value — = 1

Contains packaged clinical drug — Nexium 20 mg enteric tablet, 14, blister pack
Has pack size unit — Each
Has pack size value — = 1

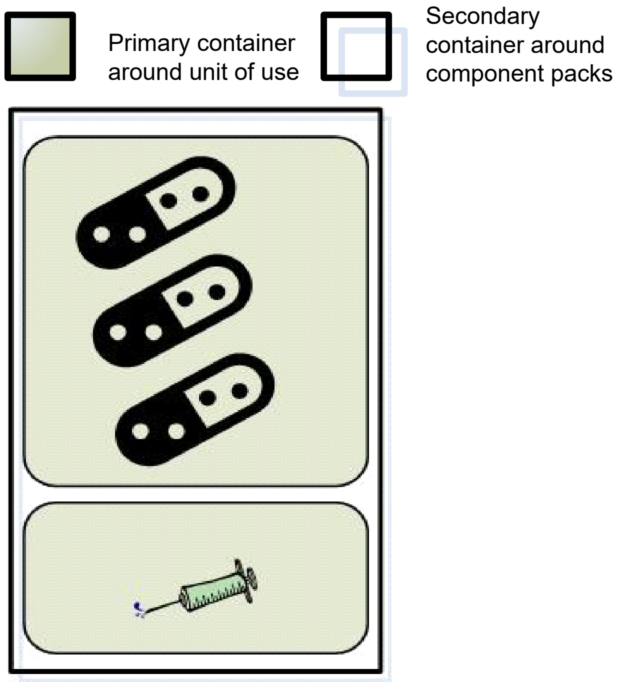
Contains packaged clinical drug — Klacid 500 mg tablet, 14, blister pack
Has pack size unit — Each
Has pack size value — = 1

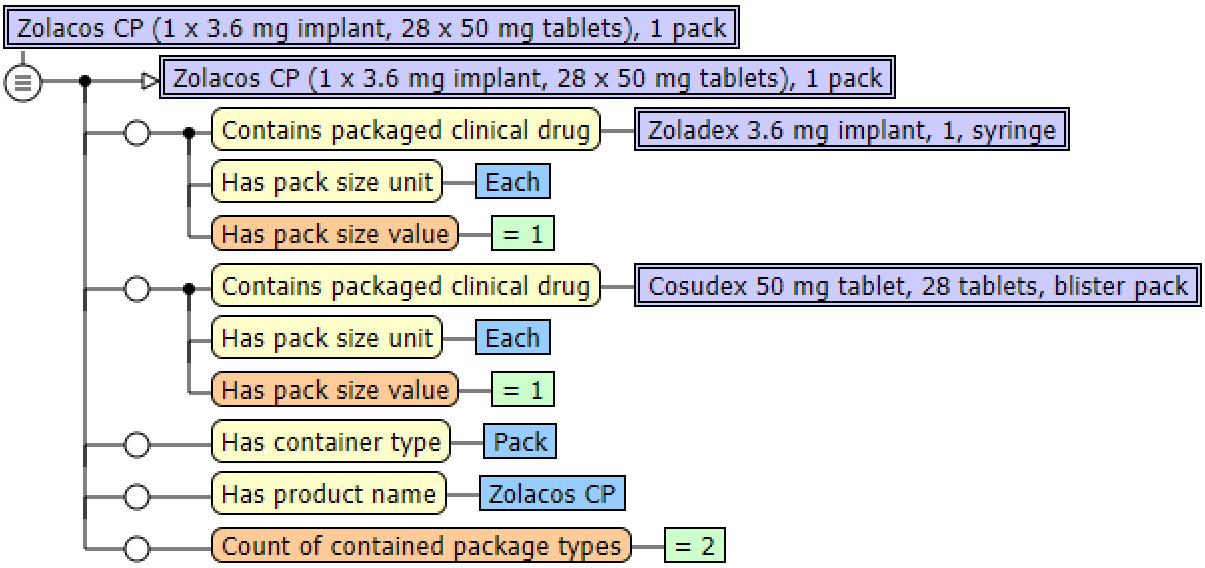
Has container type — Pack

Has product name — Nexium Hp7

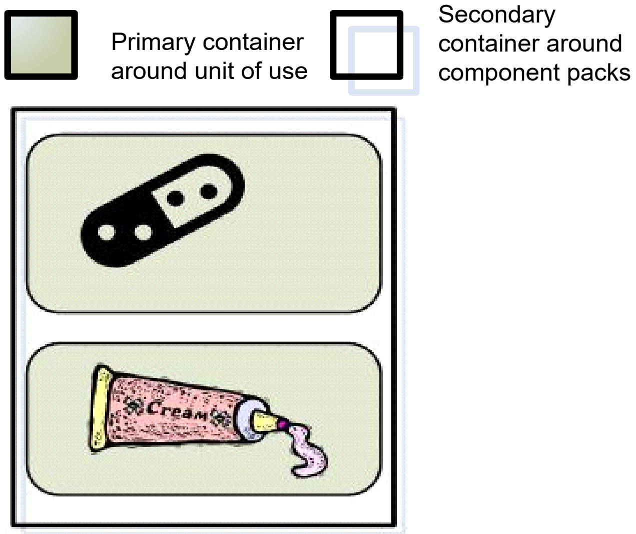
Count of contained package types — = 3

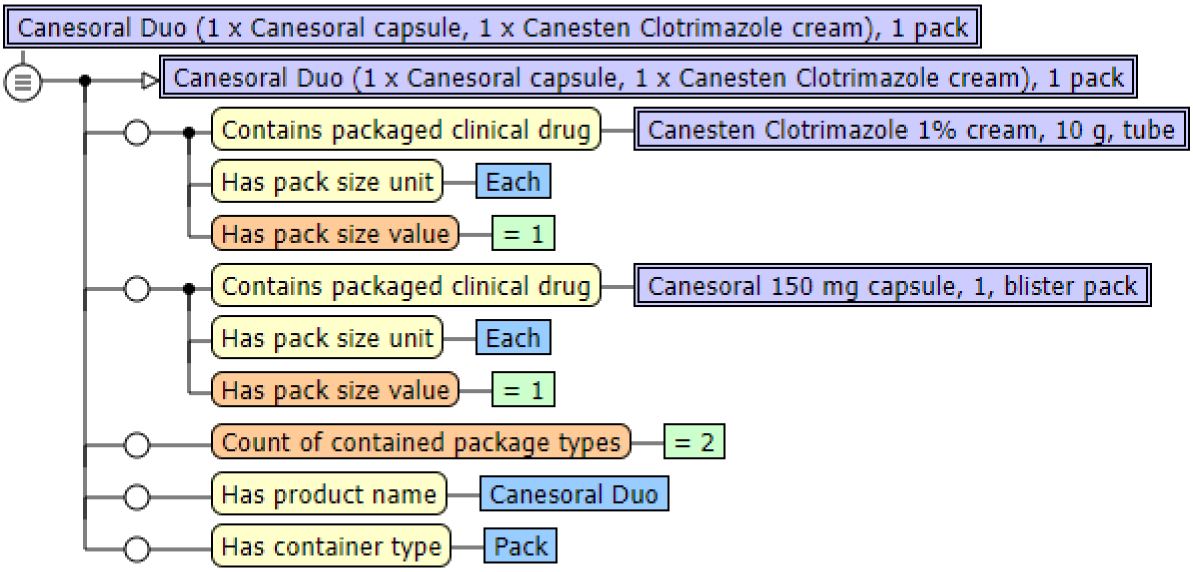
E.2.1.1 ZolaCos CP (1 x 3.6 mg implant, 28 x 50 mg tablets),1 pack


Primary container around unit of use
Secondary container around component packs


Zolacos CP (1 x 3.6 mg implant, 28 x 50 mg tablets), 1 pack
Zolacos CP (1 x 3.6 mg implant, 28 x 50 mg tablets), 1 pack
Contains packaged clinical drug
Zoladex 3.6 mg implant, 1, syringe
Has pack size unit
Each
Has pack size value
= 1
Contains packaged clinical drug
Cosudex 50 mg tablet, 28 tablets, blister pack
Has pack size unit
Each
Has pack size value
= 1
Has container type
Pack
Has product name
Zolacos CP
Count of contained package types
= 2

E.2.1.2 Canesoral Duo (1 x 150 mg capsule, 1 x 10 g cream), 1 pack


Primary container around unit of use
Secondary container around component packs
Cream


Canesoral Duo (1 x Canesoral capsule, 1 x Canesten Clotrimazole cream), 1 pack
Canesoral Duo (1 x Canesoral capsule, 1 x Canesten Clotrimazole cream), 1 pack
Contains packaged clinical drug
Canesten Clotrimazole 1% cream, 10 g, tube
Has pack size unit
Each
Has pack size value
= 1
Contains packaged clinical drug
Canesoral 150 mg capsule, 1, blister pack
Has pack size unit
Each
Has pack size value
= 1
Count of contained package types
= 2
Has product name
Canesoral Duo
Has container type
Pack

Acronyms

Acronym	Description
ADRS	Australian dialect reference set
AMT	Australian Medicines Terminology
ARTG	Australian Register of Therapeutic Goods
AUSTL	Over the counter medicines (TGA term)
AUSTR	Prescription medicines (TGA term)
BoSS	Basis of Strength Substance
CDA	Clinical Document Architecture
CR	Carriage return
CSIRO	Commonwealth Scientific and Industrial Research Organisation
CTPP	Containered Trade Product Pack
DDL	data definition language (also known as “data description language”)
FHIR®	Fast Healthcare Interoperability Resources
FSN	Fully Specified Name
GP	General Practitioner
IHTSDO	International Health Terminology Standards Development Organisation
KRSS	Knowledge Representation System Specification
LF	Line feed
MP	Medicinal Products
MPP	Medicinal Product Pack
MPUU	Medicinal Product Unit of Use
NCTS	National Clinical Terminology Service
NTS	National Terminology Server
OWL	Web Ontology Language
PBD	Pharmaceutical Benefits Division
PBS	Pharmaceutical Benefits Scheme
PT	Preferred Term

Acronym	Description
RDBMS	Relational database management systems
RF2	SNOMED CT release format 2.0
RPBS	Repatriation Schedule of Pharmaceutical Benefits
SCTID	SNOMED CT Identifier
SNOMED CT-AU	SNOMED CT, Australian release
SQL	structured query language
TGA	Therapeutic Goods Administration
TP	Trade Product
TPP	Trade Product Pack
TPUU	Trade Product Unit of Use
URI	Uniform Resource Identifier

References

Australian Digital Health Agency. (2017). *AMT Concept Model and Business Use Cases v2.1.* Retrieved from National Clinical Terminology Service Document Library: https://www.healthterminologies.gov.au/library/DH_2542_2017_AMT_Concept_Model_and_Business_Use_Cases_v2.1.pdf

Australian Digital Health Agency. (2017). *NCTS Guide for Implementers v1.0.* Retrieved from National Clinical Terminology Service Document Library: https://www.healthterminologies.gov.au/learn?content=documentlibrary

Australian Digital Health Agency. (2017). *SNOMED CT-AU Clinical Terminology Implementation Process and Checklist v1.0.* Retrieved from National Clinical Terminology Service Document Library: https://www.healthterminologies.gov.au/learn?content=documentlibrary

Australian Digital Health Agency. (2018). *NCTS Guidance for People and Processes v1.1.* Retrieved from National Clinical Terminology Service Document Library: https://www.healthterminologies.gov.au/library/DH_2757_2018_ClinicalTerminology_GuidanceforPeopleandProcesses_v1.1.pdf

Australian Digital Health Agency. (2018). *NCTS Guidance for Use in Healthcare Software v1.1.* Retrieved from National Clinical Terminology Service Document Library: https://www.healthterminologies.gov.au/library/DH_2758_2018_ClinicalTerminology_GuidanceforUseinHealthcareSoftware_v1.1.pdf

Australian Digital Health Agency. (2018). *NCTS Guidance for Use of Medical Nomenclatures in Information Exchange v1.2.* Retrieved from National Clinical Terminology Service Document Library: https://www.healthterminologies.gov.au/library/DH_2759_2018_ClinicalTerminology_GuidanceforUseofMedicalNomenclaturesinInformationExchange_v1.2.pdf

Australian Digital Health Agency. (2019). *SNOMED CT-AU Development Approach for Reference Sets v3.3.* Retrieved from National Clinical Terminology Service Document Library: https://www.healthterminologies.gov.au/learn?content=documentlibrary

Australian Digital Health Agency. (2020). *AMT Editorial Rules v2.7.* Retrieved from National Clinical Terminology Service Document Library: https://www.healthterminologies.gov.au/learn?content=documentlibrary

Australian Digital Health Agency. (2022). *SNOMED CT-AU Mapping Guidelines v3.0.* Retrieved from National Clinical Terminology Service Document Library: https://www.healthterminologies.gov.au/library/DH_3643_2022_SNOMEDCTAU_MappingGuidelines_v3.0.pdf

Australian Digital Health Agency. (2024). *AMT v4 Editorial and Naming Conventions.* Retrieved from https://www.healthterminologies.gov.au/document-library/

Australian Digital Health Agency. (2024). *AMT v4 Model Specification.* Retrieved 2024, from https://www.healthterminologies.gov.au/document-library/

SNOMED International. (n.d.). SNOMED CT Editorial Guide. Retrieved June 2020, from http://snomed.org/eg

SNOMED International. (n.d.). SNOMED CT Release File Specifications. Retrieved June 2024, from http://snomed.org/rfs

SNOMED International. (n.d.). SNOMED CT Terminology Services Guide. Retrieved June 2024, from http://snomed.org/tsg

SNOMED International. (n.d.). SNOMED Glossary. Retrieved June 2020, from http://snomed.org/gl

SNOMED International. (n.d.). Technical Implementation Guide. Retrieved June 2020, from http://snomed.org/tig

SNOMED International. (n.d.). URI Standard. Retrieved June 2020, from http://snomed.org/uri