nehta

# SNOMED CT-AU Release 20141130
# Australian Implementation Guidance

30 November 2014

Approved for external use

Document ID: NEHTA-1836:2014

**National E-Health Transition Authority**

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



**Acknowledgement**

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

# Document information

## Key information

| | |
|---|---|
| **Owner** | Head of National Services Operation and Management |
| **Date of next review** | 15 April 2015 |
| **Contact for enquiries** | NEHTA Help Centre<br>t: 1300 901 001<br>e: help@nehta.gov.au |

## Product version history

| Product version | Date | Release comments |
|---|---|---|
| 20131130 | 30 Nov 2013 | Initial Release. |
| 20140531 | 31 May 2014 | Updated for SNOMED CT-AU May 2014 Release.<br>Reference set Implementation Guide integrated. |
| 20141130 | 30 Nov 2014 | Updated for SNOMED CT-AU November 2014 Release. |

# Table of contents

# 1 Introduction

## 1.1 Purpose

This document provides introductory implementation guidance for software developers, technical consumers and the general SNOMED CT®[1]-AU community of practice. It is intended to provide practical guidance for Australian implementations and although it provides introductory details regarding the terminology, it is to be used in conjunction with the IHTSDO's *SNOMED CT Technical Implementation Guide* [1].

The implementation approaches detailed here are for guidance only and are provided with the assumption that each implementation will involve varying technical capabilities that may affect how the terminology is implemented. Feedback is actively encouraged and the document will be updated and revised so that it meets its objective of providing practical advice for implementers.

## 1.2 Intended audience

This document is intended for those in the SNOMED CT-AU community of practice who have some understanding of SNOMED CT, particularly for vendors of technical healthcare products or systems. This includes developers and testers who are responsible for producing, assuring or maintaining products that integrate with SNOMED CT-AU. The document assumes the reader has a basic understanding of software development and database management.

## 1.3 Scope

This document is limited to providing detailed implementation information for Australian implementers that is not addressed within the *SNOMED CT Technical Implementation Guide* [1]. It is limited to information about how to implement, rather than information on the development of release artefacts or products. For example, the NCTIS *Reference Set Library* [2]and *SNOMED CT-AU Development Approach for Reference Sets* [3] describe products that are part of the release whereas this document is intended for guidance on the options to implement certain aspects of SNOMED CT-AU.

Neither does it cover content already described in the *SNOMED CT Technical Implementation Guide* to any significant depth. Appropriate chapters will be cross-referenced from within this document, where further reading is recommended. All code samples provided are for demonstration purposes only, and do not represent the most efficient or robust approach. Implementers are advised to conduct their own performance tuning and ensure appropriate exception handling.

## 1.4 SQL examples

Where possible, example SQL scripts are provided to assist in demonstration of how features work. The schema for these code samples is aligned with the guide included in the *SNOMED CT-AU Reference Set Implementation Toolkit* [4].[2] If the

[1] IHTSDO®, SNOMED® and SNOMED CT® are registered trademarks of the IHTSDO.
[2] Hereafter: "reference set toolkit". This toolkit consists of a java/C# app, a database, and an accompanying guide. Once the user has set up and compiled the application, the database created can be used to execute the queries.

reference set toolkit has also been set up, most of the queries provided may be executed, as provided, on the same database.

It should also be noted that the reference set toolkit uses a Snapshot release, which contains active and inactive content. However, most queries in this document do not differentiate between active and inactive content, for brevity. Developers who wish to exclude inactive components from their queries should generally apply the “active=1” clause to each table.

# 2 SNOMED CT Australian Release

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

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.

| | |
|---|---|
| **Concepts table** | Holds the clinical concepts that make up SNOMED CT. A concept is given meaning by its Fully Specified Name, which is held in 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 table** | 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 table** | Provides information about the relationships between the concepts. These relationships define and bring meaning to the individual concepts relative to other concepts. |

The SNOMED CT concept model provides 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.

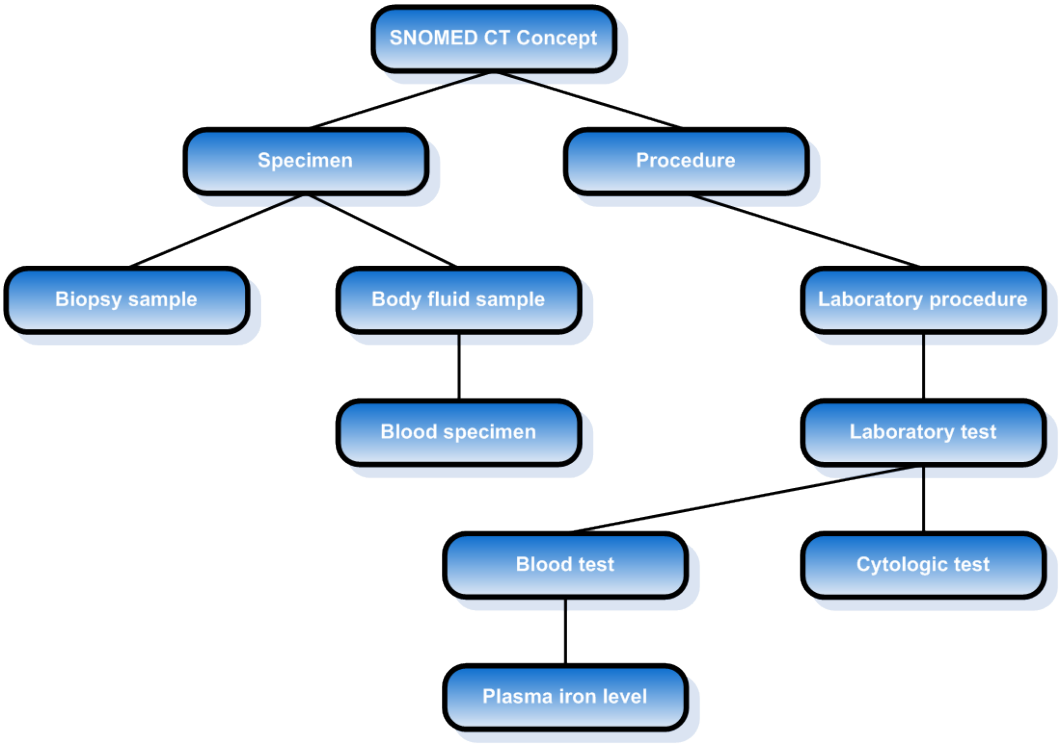


*Figure 1: A conceptual view of the SNOMED CT hierarchy*

SNOMED CT also provides a mechanism for extending these core files through the use of reference sets. Reference sets can be used for any number of purposes such as annotating content with additional information or identifying subsets for a specific purpose. Detailed specifications for the various types of reference sets are available in the *SNOMED CT Technical Implementation Guide* [1], however Section 3 provides additional information for reference sets specific to SNOMED CT-AU.

The relationship of the International Release of SNOMED CT to the content developed for Australia is presented in the figure below. It shows how SNOMED CT-AU comprises the four core tables of SNOMED CT together with the additional Australian content.

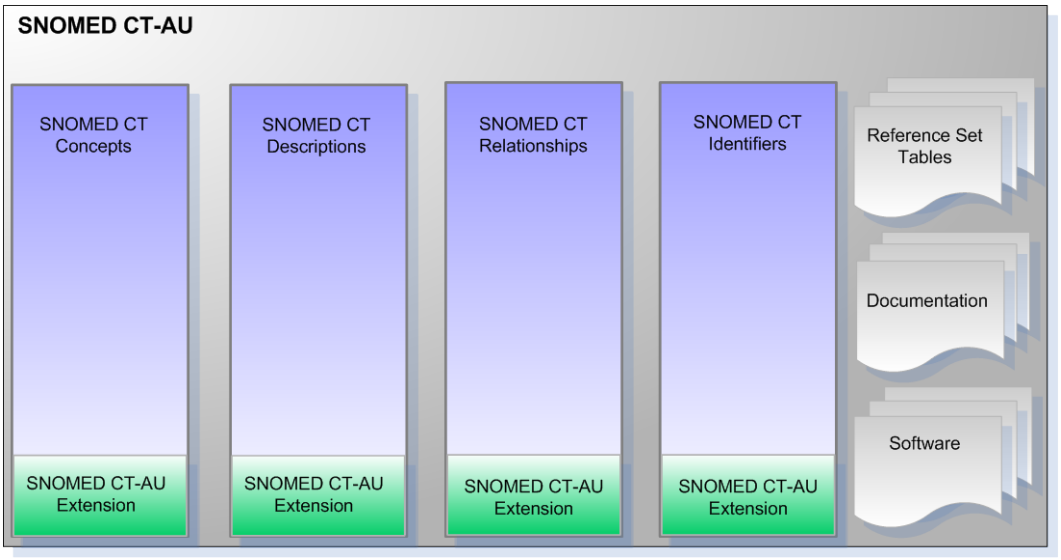


*Figure 2: Schematic representation of SNOMED CT-AU*

SNOMED CT member countries are allocated unique namespace identifiers under which they may create new SNOMED CT identifiers for extension content. All new content developed by NEHTA has been assigned a SNOMED CT identifier from the Australian namespace.

If any of the proposed concepts for the Australian namespace are applicable to a wider domain, they may be promoted to the International Release of SNOMED CT. Licence holders may apply for an affiliate namespace through NEHTA.

## 2.1 Release format

SNOMED CT-AU is released exclusively in Release Format 2 (RF2). The original release format (now referred to as RF1) will eventually be deprecated and is not supported for use in Australia. For detailed technical specifications and guidance on concerning Release Format 2, please consult the *SNOMED CT Technical Implementation Guide* [1]

Not all of the international reference sets described here are included in the SNOMED CT-AU release. The applicability of the internationally available reference sets will be periodically assessed and may be subject to change over time.

# 3 SNOMED CT-AU reference sets

### 3.1.1 Purpose and definition

SNOMED CT-AU has a large range of content covering various health care disciplines and clinical specialties. Reference set data structures provide the foundation pieces of the generic extensibility mechanism in RF2 *SNOMED CT Technical Implementation Guide* [1]. 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 useable groups. Reference sets can be used for many different purposes:

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

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

#### 3.1.1.1 Structural reference sets

Structural reference sets may be required in implementations to fulfil technical requirements. Structural reference sets describe:

- other, critical, details of SNOMED CT content not detailed in the core tables (e.g. the *Australian dialect reference set*); and
- the technical metadata associated for the release (e.g. Module dependencies and reference set descriptors).

#### 3.1.1.2 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. The 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.

#### 3.1.1.3 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. Unlike bound reference sets, however, they do not take into account any other definitions or data items that may co-exist where these reference sets might be implemented.

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

Reference sets with specific bindings described by the National Clinical Terminology and Information Service (NCTIS) are categorised according to those bound to NEHTA Clinical Information Specification archetypes (or data groups) and those bound to other clinical information specifications.

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

### 3.1.2 Reference sets produced

The reference sets developed and released by the NCTIS will be of one of the types described below.

#### 3.1.2.1 Foundation reference sets

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

#### 3.1.2.2 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 (e.g. terminology commonly used to describe patients admitted to an orthopaedic ward). Intermediate reference sets use the same approach and will contain more focused content than a Broad context reference set.

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

#### 3.1.2.3 Specific reference sets

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

#### 3.1.2.4 Structural reference sets

A suite of Structural reference sets will be developed as a mechanism for managing SNOMED CT-AU as an extension, its inherent data structures, and required release formats.

## 3.2 Release cycle

SNOMED CT-AU will be released and maintained by NEHTA. The IHTSDO releases the SNOMED CT International Release every six months. NEHTA will update reference sets based on the new International Release as soon as practicable after the International Release has been analysed and the new reference sets can be generated. Additional reference sets developed by NEHTA will also be added to releases as they become available. Alterations and refinements will also be made based on feedback from implementations.

## 3.3 Distribution format and file naming

### 3.3.1 Distribution format

The SNOMED CT-AU release is distributed in a single zip archive, named “SNOMED CT-AU Terminology Release File Bundle.zip”. This archive contains the following components:

- Documentation
- RF2 Release
  - Delta
  - Full
  - Snapshot

### 3.3.2 File naming conventions

See *SNOMED CT Technical Implementation Guide* [1] for further details.

## 3.4 Maintenance and support

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

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

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

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

# 4 Implementing SNOMED CT-AU

## 4.1 General implementation considerations

The path to take when developing applications using SNOMED CT-AU will be unique for each implementation. Choosing the best path will require the implementer to first perform sufficient analysis and design to extract the required SNOMED CT-AU data and store it in a form suitable for the application's or system's needs.

There are two generic activities that must be performed as part of all implementations; namely, importing the required terminology files, and then incorporating the terminology into the application functions.

It is critical that before a reference set is used in an implementation, the implementer fully understands the scope and context within which the reference set is intended for use. This information will be provided as part of the release documentation of any NEHTA-produced reference set. The scope and type of the reference set are key parts of the implementer's analysis, which in turn helps to ensure that the required attributes will be correctly imported.

It is expected that an implementer will wish to transform the provided reference set into something more suited to their environment, e.g. a relational database or Microsoft Excel spreadsheet.

Reference sets may sometimes be useful in providing constrained lists of values that could be used to populate user interfaces. Typically, to achieve this it will be necessary to determine the Preferred Term for the given member concept from the language dialect reference set, and use that term to populate the user interface text.

SNOMED CT-AU (including reference sets) is not a static terminology; it will be further developed and maintained. It has a change history tracking mechanism that appends the changed row to the file, which must be given consideration to ensure that implementations are capable of handling such changes.

The extent of the data to be imported is governed by the requirements and design of the incorporating system; hence such decisions are left to the implementer.

### 4.1.1 Analysis, transformation, and importing of SNOMED CT-AU terminology and reference sets

There are two general mechanisms for extracting and importing SNOMED CT-AU data:

1 The relevant SNOMED CT-AU data can be imported into structures within the host system that are similar to the SNOMED CT-AU release format.

2 The required data can be generated as simple lists and imported into the required host data structures.

There will, however, be other approaches that involve both mechanisms, depending upon local system requirements and the implementation path chosen.

Note: The *Australian dialect reference set* must be selected to determine the Preferred Term for each concept. See Section 4.4.2 for further details.

See the *SNOMED CT Technical Implementation Guide* [1] for guidance on importing SNOMED CT into applications.

## 4.2 Data types

Full details of the data types used in the release files are available in the *SNOMED CT Technical Implementation Guide*.

## 4.3 File types

The distribution files for reference sets:

- are tab-delimited text files;
- are UTF-8 encoded;
- contain a column header row; and
- use DOS-style line termination (i.e. lines are terminated with a carriage return character followed by a line feed character).

## 4.4 Reference set patterns

All reference sets are based on a generic data structure that is able to be extended to meet application requirements. The first four fields fulfil the same purpose as those in the core Concept, Description, and Relationship files: see the *SNOMED CT Technical Implementation Guide* for related information.

*Table 1: 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 part of. It should be a descendant of *Reference set (Foundation metadata concept)*. This can be found in the *SNOMED CT Model Component (metadata)* hierarchy. |
| referencedComponentId | Uniquely identifies the component that this row relates to, thus defining membership of this component in the reference set. |
| Zero or more other fields | Optional fields. |

The first six fields above are used in all reference sets with additional optional fields supporting either SCTID, String, or Integer data types, appended as required. Different reference set patterns can be created by adding varying combinations of data type fields to this basic structure. The *SNOMED CT Technical Implementation Guide* [1] details a number of predefined patterns. Over time, NEHTA will release various types (patterns) of reference sets. As they are introduced, this document will be extended to describe them.

### 4.4.1 Simple type reference set

This reference set pattern is just the basic reference set member format (see Table 1) with no additional fields. It is used to identify a subset of content. All clinical reference sets currently released by NEHTA fall under this category. All current SNOMED CT-AU simple type reference sets identify a set of ConceptIds. Most of the clinical reference sets contained in this release resemble the example below.

*Table 2: Example usage of an attribute value reference set*

| Common fields … | refSetId | referencedComponentId |
|---|---|---|
| … | *"Specimen type reference set"* | *"Urine specimen"* |
| … | *"Specimen type reference set"* | *"Sputum specimen"* |
| … | *"Specimen type reference set"* | *"Sweat specimen"* |

### 4.4.2 Language type reference sets

This reference set pattern supports expression of preference and acceptability of descriptions for specific dialects of a language or relevant to a particular context. The *Australian dialect reference* set is a Language type reference set provided with SNOMED CT-AU that specifies general context Australian acceptability and preference for English synonyms. Language type reference sets are considered to be supporting reference sets as they are critical to technical implementation, but do not necessarily have any clinical constraints. The following table describes the fields within a Language type reference set file.[3]

*Table 3: Language type reference set format*

| Field | Purpose |
|---|---|
| Id | As per Table 1. |
| effectiveTime | As per Table 1. |
| Active | As per Table 1. |
| moduleId | As per Table 1. |
| refSetId | Set to the concept ID for the reference set. This identifies the dialect or context for which the Language type reference set is relevant. |
| referencedComponentId | Identifies the description from the description table to be referenced. |
| valueId | Identifies the relevant description type for the referenced description in this dialect. Can take values of "Preferred" or "Acceptable". |

A Language type reference set states whether a particular description should be used as the Preferred Term or is otherwise acceptable for use. The referencedComponentId will be a description ID, and the valueId field will be a concept designating the description as "Preferred" or "Acceptable". Any descriptions not referenced should be regarded as "not acceptable" in the context of the relevant dialect.

[3] Adapted from Section 5.6.2.8 "Language Reference Set" of *SNOMED CT Technical Implementation Guide* [1].

Table 4 below illustrates a sample of how a Language type reference set may appear. Note that the fields shown would only contain SctIds (although text descriptions have been provided to assist readability). The “common fields” (see Table 1) are not detailed.

*Table 4: Example usage of a Language type reference set*

| **Common fields …** | **refSetId** | **referencedComponentId** | **valueId** |
|---|---|---|---|
| … | “Australian English” | *Structure of cerebellar biventral lobule* | “Preferred” |
| … | “Australian English” | *Cruciate lobe* | “Acceptable” |

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

Implementation of the ADRS is required, so that the Preferred Term for each concept may be identified. Additional technical guidance on implementing the ADRS is available in Section 6.

## 4.5 Validation

Validating the integrity of a reference set must be performed prior to its use within applications or systems. Further maintenance validations are needed from that point onward.

The level and type of validation required will vary depending upon the implementation; however, the main checks necessary are:

- That the base reference set has not been corrupted in any way. This may simply be a check that all SCTIDs are valid.
- That all required data for reference set members and components are available within their system or application.
- That the production system is version-aware, given that reference sets will be maintained and updated. This could include the addition, retirement, or modification of reference set members.

## 4.6 Updates and maintenance

It is likely that most reference sets will be updated, some more regularly than others, depending upon their subject matter and scope. The implementer should take this into consideration when building import and data transform processes, or applications that use this data. Processes must be adapted to ensure that backwards compatibility between versions can be achieved in their system. Each SNOMED CT-AU release will include the updated reference sets and will be identified by SNOMED CT’s date-based versioning system. Concepts may be retired from

reference sets but neither the concept itself nor its historical membership will be deleted.

Work is underway to add Australian content where SNOMED CT is inadequate for Australian healthcare requirements.

Where a system may receive messages from other systems, implementers should also build capacity for handling unexpected codes. Unexpected codes may be encountered when:

- The sender has a more recent version of SNOMED CT-AU than the recipient, and is therefore able to send new codes of which the receiving system is not yet aware.
- The sender has an outdated version of SNOMED CT-AU and is sending codes that have since been deprecated in the version used by the recipient. However, if the recipient system is *history aware*, the inactive code should still be recognised.
- The sender has created a local extension, not available to the recipient, and therefore unrecognised.

# 5 Scenarios for implementation

This section discusses three generic scenarios for SNOMED CT-AU implementations. Although there are many more possible scenarios, the principles described here should be generally applicable.

## 5.1 Existing system with non-SNOMED CT local terminology in need of external interoperability

A common scenario will be one where systems that are using terminologies other than SNOMED CT require interoperability with external SNOMED CT-based systems.

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

In order 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 SNOMED CT distribution files. For each reference set, a file needs to be created with the following fields:

- Concept ID
- Preferred Term description text

This could be achieved by using SQL scripts to create a join between the Descriptions Table, each of the reference set files, and the applicable language set. The ADRS indicates the general language preferences for Australian implementations.

The resulting file now has 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:

- Concept Id
- Reference set Id
- Preferred Term description text
- 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, we recommend including the “Effective date” field in the mapped file. This will provide traceable and reproducible message translation if the effective date is used in queries against the map during the

translation process. Also, updates to the maps can be achieved by simply adding new rows to the table(s) used to store the mapped data.

See Section 10 for more details on mapping.

## 5.2 Development of a new system using SNOMED CT-AU

As new systems are developed, there will be the ability to use SNOMED CT-AU as the system's native coding system, making interoperability easier. To implement SNOMED CT-AU in a new system, it will be necessary to refer to the *SNOMED CT Technical Implementation Guide* [1] and to utilise the advice given. 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.

## 5.3 Using SNOMED CT-AU as an interface terminology

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

Analysis must be undertaken by the implementer 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” (display text) for each concept must also be determined. In most instances, it is strongly recommended that the “Preferred Term” be used. Clearly, however, there are other descriptions such as Synonyms that could also be used as display text.

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

# 6 Australian dialect reference set

The *Australian dialect reference set* (ADRS) is a Language type reference set. Language type reference sets are used to express local language preferences and dialectic differences across descriptions from a parent language; for example British English preferences versus United States English preferences.

SNOMED CT-AU provides two types of descriptions:

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

Language type reference sets in RF2 annotate RF2 Synonym descriptions with one of three mutually exclusive values to provide localisation:

**Preferred** Indicates that Synonyms annotated with this value are the preferred way of describing the concept (also known as the "Preferred Term"). A Language type reference set is required to have exactly one active "preferred" Synonym for each concept.

**Acceptable** Indicates that Synonyms annotated with this value are not the preferred way of describing a concept, but are acceptable in the Language type reference set's context. A Language type reference set may refer to zero or more of a concept's descriptions with this value.

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

## 6.1 Technical summary

The ADRS is a component reference set as described in Section 5.6.2.8 "Language Reference Set" of the *SNOMED CT Technical Implementation Guide* [1]. 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.

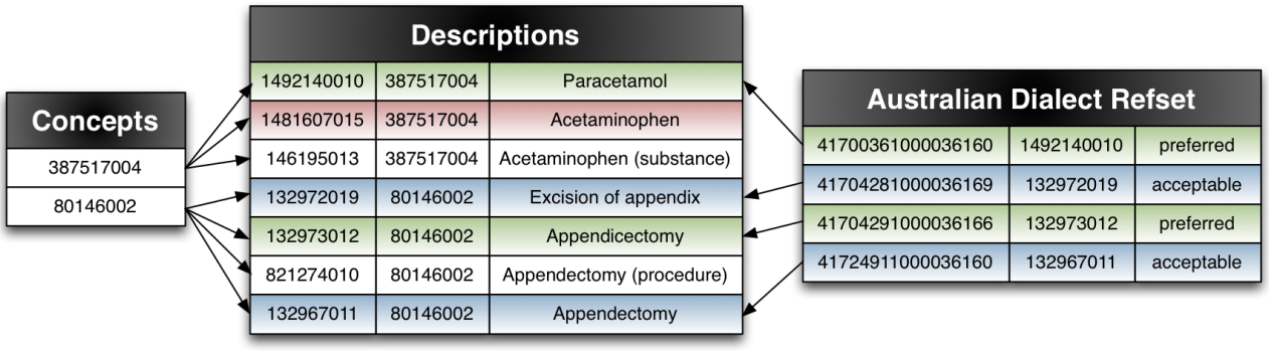


*Figure 3: Relationship between the Concepts, Descriptions and Language type reference set*

Figure 3 above shows an example taken from SNOMED CT-AU, showing how the *Australian dialect reference set* expresses that:

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

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

Note: Fully Specified Names are not referenced – they are always acceptable and preference is irrelevant.

Note: Columns have been omitted from the example for brevity.

### 6.1.1 SQL examples

Consider the conceptId 387517004. All available descriptions can be retrieved using the following query:

```
select descriptions.conceptId,descriptions.id,term
from descriptions
where descriptions.conceptId = 387517004
```

The descriptions that are referenced in the ADRS can be identified by extending the query as:

```
select descriptions.conceptId,descriptions.id,term, valueid
from descriptions
inner join description_refset AS ADRS
on descriptions.id = ADRS.referenceddescriptionid
where descriptions.conceptId = 387517004
```

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

```
select descriptions.conceptId,descriptions.id,term, valueid
from descriptions
inner join description_refset AS ADRS
on descriptions.id = ADRS.referenceddescriptionid
where descriptions.conceptId = 387517004
and ADRS.valueid = 900000000000548007 -- ConceptId for 'Preferred'
and ADRS.active = 1
```

Note: It is necessary to apply the “ADRS.active = 1” criterion 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_ADRS_PreferredTerm
//
CREATE FUNCTION get_ADRS_PreferredTerm(candidate bigint(20)) RETURNS
varchar(300)

BEGIN
   RETURN (select term
                from descriptions
                inner join description_refset AS ADRS
                on descriptions.id = ADRS.referenceddescriptionid
                where descriptions.conceptId = candidate
                and ADRS.valueid = 900000000000548007
                and ADRS.active = 1);
END
//
```

This function can then be called whenever a Preferred Term is required.

```
select get_ADRS_PreferredTerm(64459004);
```

## 6.2 Implementation suggestions

### 6.2.1 Performance

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

Alternatively, index optimisation alone may provide sufficient performance gains.

# 7 Term searching

Searching for appropriate SNOMED CT-AU concepts by entering some text and selecting a concept from search results is the most common method for giving end users access to the broad range of content in SNOMED CT-AU. To ensure a positive user experience, it is important to provide an effective search capacity. Section 7.6.1 “Text Searches” of the *SNOMED CT Technical Implementation Guide* [1] provides some options for producing effective searches, and some additional approaches are described below.

## 7.1 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”.

Using the following query produces the expected results and with acceptable performance:

```
select term from descriptions
where term like 'frac%';
```

However, such “starts with” searches will not identify terms where “frac” appears anywhere but the start of the description. It is possible to prefix the search string with a wild card such that:

```
select term from descriptions
where term like '%frac%';
```

This query will yield results that contain “frac” anywhere within the string, however since basic SQL indexing typically relies on the characters at the front of a string – the index is useless and performance will suffer.

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

- **Native Full Text Indexing:** Most relational database management systems now provide native Full Text indexing. The features and syntax vary between systems, so the relevant Relational Database Management System user documentation should be consulted. Such indexes require little extra setup and are easy to use.
- **Specialised Software Library**: 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[4] is an example of such a library, and versions are available for most major development languages.

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

## 7.2 Restricting scope to reference sets

An unrestricted search upon all of SNOMED CT-AU 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 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* [5] for further details.

One way to restrict the scope is to limit the search to concepts from a reference set 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. Similarly, certain data elements in NCTIS information models have value domains that are restricted to particular reference sets. Implementers must ensure that users only populate these fields using the relevant concepts.

### 7.2.1 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
where term like 'ulcer';
```

As noted this would yield two results. To restrict the search to *Clinical findings*, use the reference set 32570071000036102|*Clinical finding foundation reference set* |, as follows.

```
Select term from descriptions
where term like 'ulcer'
and conceptId in (select referencedConceptId from concept_refset
                  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 7.3 below.

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

## 7.3 Use all Synonyms but limit to Preferred Terms

The availability of Synonyms in SNOMED CT-AU 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

"raised blood pressure" may return two concepts with that Synonym, both from the *Clinical findings* hierarchy:

- 24184005 |*Raised blood pressure*|
- 38341003 |*Raised blood pressure*|

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*|), but from a user perspective, this can raise confusion or doubt relating 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 concept ID), rather than displaying only the Preferred Terms for those concepts.

### 7.3.1 SQL examples

The simplest way to produce a list showing only individual concepts and Preferred Terms is to use a function such as get_ADRS_PreferredTerm(conceptId), as described in Section 6.1.1.

Re-using the example from Section 7.3, searching for "raised blood pressure" would result in unconstrained queries that might look like:

```
select conceptId, term from concepts
inner join descriptions on concepts.id = descriptions.conceptid
where term = 'raised blood pressure'
and concepts.active = 1
and descriptions.active = 1;
```

This returns the following:

- 24184005 |*Raised blood pressure*|
- 38341003 |*Raised blood pressure*|

We can change the query so that the result set provides the Preferred Terms, by using the created function (bolded):

```
select conceptId, get_ADRS_PreferredTerm(conceptId) from concepts
inner join descriptions on concepts.id = descriptions.conceptid
where term = 'raised blood pressure'
and concepts.active = 1;
```

This revised query will now return the following results:

- 24184005 |*Finding of increased blood pressure*|
- 38341003 |*Hypertension*|

Note that the same concepts are returned, but they can now be distinguished by an end user (who would 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_ADRS_PreferredTerm(conceptId)
from concepts
inner join descriptions on concepts.id = descriptions.conceptid
where term like '%hypertension%'
and concepts.active = 1;
```

This query returns 201 unique concepts, based on 517 matching terms[5].

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.

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

# 8 Subsumption queries

The concepts in SNOMED CT-AU are organised in hierarchy of increasing specificity, where concepts at the top are broad classes, and those at the bottom representing more specific concepts. Specificity increases through the tree, with concepts joined by 116680003 |*Is a*| relationships being more specific.

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

Section 7.7 “Testing and traversing subtype relationships” of the *SNOMED CT Technical Implementation Guide* [1] describes many aspects of using these relationships to determine if one concept is a subtype of another. Of specific interest is the use of a transitive closure table, as described in Section 7.7.5.2 “Transitive Closure Implementation”, which also includes code for producing a transitive closure table (though the table names will need to be modified if used in the Reference Set Toolkit database). The transitive closure table provides a fast and easy way for determining if two concepts have a subtype/supertype relationship.

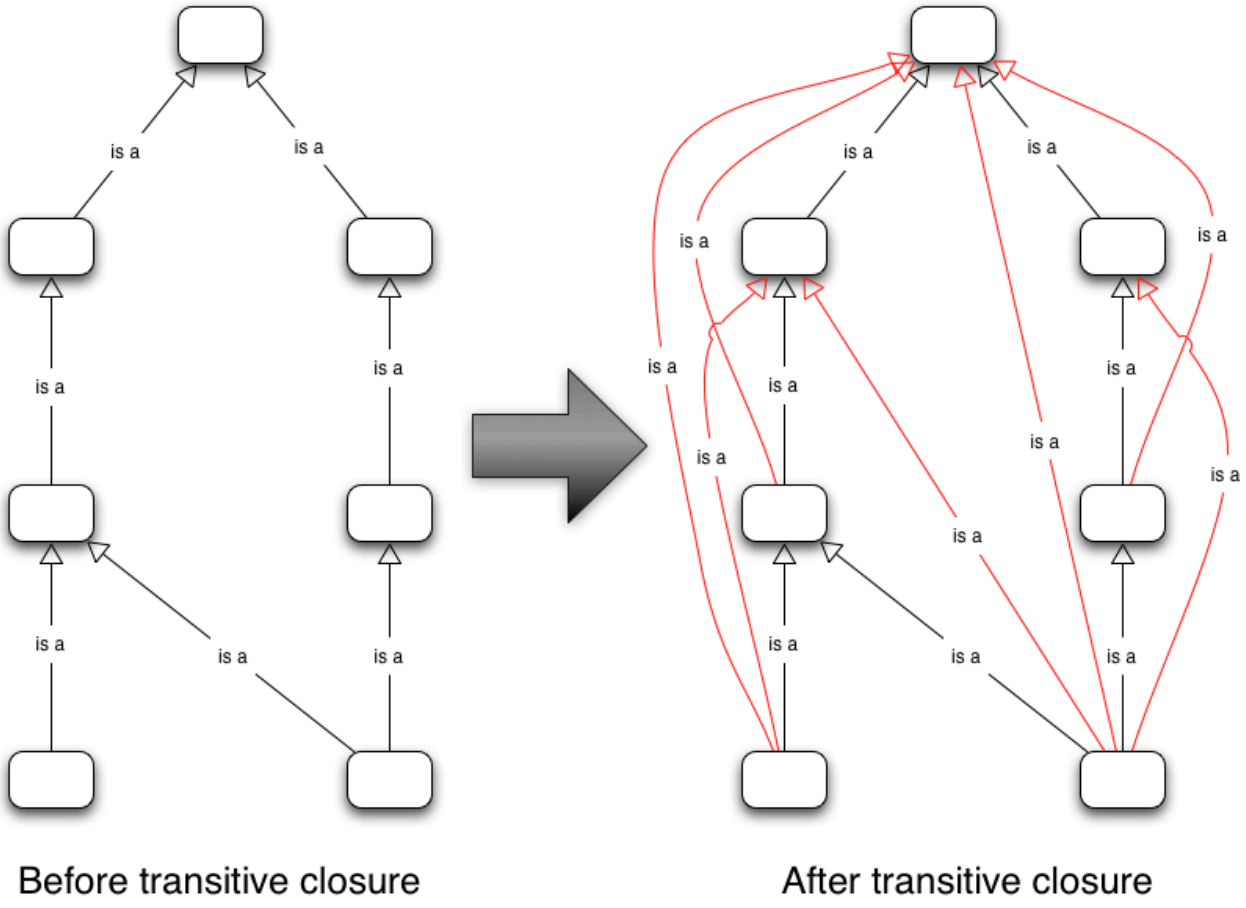


*Figure 4 Example of a transitive closure*

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 specific to the Reference set tool kit is provided in Appendix A.1.

## 8.1 SQL examples

### 8.1.1 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
where typeId = 116680003
and sourceId = 442183006
```

This query shows the two IS A relationships *Psychogenic dyskinesia* has.

*Table 5: Distributed IS A relationships for Psychogenic dyskinesia*

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

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

```
select sourceId,destinationId from transitiveclosure
where sourceId = 442183006
```

The query results are tabulated below. Note that the transitive closure query does not specify a relationship typeId, as all relationships are of the type 116680003|Is a| in the transitive closure table.

*Table 6: 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* |

| sourceId | destinationId |
|---|---|
| *Psychogenic dyskinesia* | *Psychological finding* |
| *Psychogenic dyskinesia* | *Functional finding* |
| *Psychogenic dyskinesia* | *Finding by site* |
| *Psychogenic dyskinesia* | *SNOMED CT Concept* |
| *Psychogenic dyskinesia* | *Psychological finding of perception* |
| *Psychogenic dyskinesia* | *Clinical history and observation findings* |
| *Psychogenic dyskinesia* | *Finding of movement* |
| *Psychogenic dyskinesia* | *Mental state, behaviour and/or psychosocial function finding* |
| *Psychogenic dyskinesia* | *Clinical finding* |

### 8.1.2 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 8.1.1. Consider the concept 54556006 |*Fracture of ulna*|; 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 TransitiveClosure
where destinationId = 54556006
```

This query reveals the 45 concepts that are subtypes of this concept including the seven immediate children[6].

*Table 7: Descendants of "Fracture of ulna"*

| Immediate Children | All Descendants |
|---|---|
| Birth fracture of ulna | Birth fracture of ulna |
| Closed fracture of ulna | Closed fracture of ulna |
| Fracture of distal end of ulna | Fracture of distal end of ulna |
| Fracture of proximal end of ulna | Fracture of proximal end of ulna |
| Fracture of radius AND ulna | Fracture of radius AND ulna |
| Fracture of shaft of ulna | Fracture of shaft of ulna |
| Open fracture of ulna | Open fracture of ulna |
| | Open fracture of distal end of ulna |
| | Closed fracture of proximal end of ulna |
| | Closed fracture of distal end of ulna |
| | Monteggia's fracture |
| | Fracture of olecranon |

[6] As of SNOMED CT-AU May 2014. Specific numbers may vary for other releases.

| Immediate Children | All Descendants |
|---|---|
| | Fracture of coronoid process of ulna |
| | Fracture of ulnar styloid |
| | *… and 31more…* |

### 8.1.3 Retrieving complex sets of descendants 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 greater set of concepts. As an example, there are 288 399907009 |*Animal bite wound*| concepts. Amongst these are 172 409985002 |*Arthropod bite wound*| concepts.

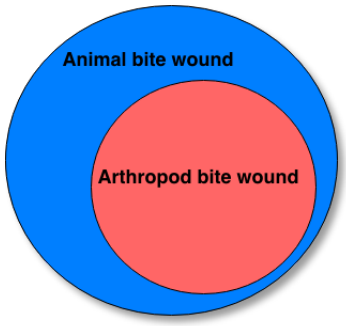


*Figure 5: Relationship between Animal and Arthropod bite concepts*

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

```
select sourceId from transitiveclosure
where destinationId = 399907009 -- Animal bite wound
-- exclude the concepts that are 409985002|Arthropod bite wound| descendants
and sourceId not in (select sourceId from transitiveclosure
                     where destinationId = 409985002
                     union select 409985002 -- and "Arthropod bite wound"
                     );
```

This query provides the set of 115 concepts that are not *Arthropod bite wounds*.

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

### 8.1.4 Retrieving the intersection of two sub-hierarchies

Concepts in SNOMED CT-AU are organised in a polyhierarchy[7], 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.

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

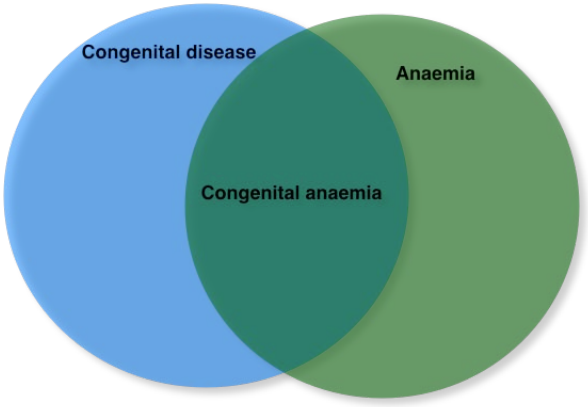


*Figure 6: Intersection of Congenital diseases and Anaemias*

Identifying the intersection is an excellent application for using an inner join query as illustrated by the following query:

```
select anaemia.sourceId from transitiveclosure as anaemia
   inner join transitiveclosure as congenital
   on anaemia.sourceId = congenital.sourceId
   where anaemia.destinationId = 271737000 -- | anaemia |
   and congenital.destinationId = 66091009; -- | congenital disease |
```

Alternatively, a similar query to that described in Section 8.1.3.

```
select sourceId from transitiveclosure as anaemia
where anaemia.destinationId = 271737000 -- | anaemia |
and sourceId in (
         select sourceId from transitiveclosure as congenital
         where congenital.destinationId = 66091009 -- | congenital disease |
               );
```

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

# 9 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 where exchange message value domains involved. In particular:

- A custom subset of an existing reference set will still allow the sending of compliant messages. However, incoming messages potentially carry codes not in the custom reference set.
- A custom reference that contains additional concepts to those specified in national exchange specifications, should not include those codes in messages.

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 (e.g. “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 use and maintenance perspective, given that it is highly likely that the original reference set will be updated by NEHTA.

Where a copy of the reference set is created and customised to satisfy local needs, validation and quality assurance would also need to be conducted locally.

Altering the contents of a specific reference set may invalidate it against its original scope. Sufficient analysis should be done to fully understand the impact of any proposed changes.

- **Maintenance and Update** – Reference sets will likely be subject to updates; hence it is advised that any alterations be reproducible against a newly released version of the original reference set. Reference sets developed by NEHTA will be updated to reflect any changes in the six-monthly International Release of SNOMED CT.
- **Quality and Safety** – Quality and safety implications of any changes should be considered and careful analysis of these implications undertaken.

In creating these there are two broad approaches: Inclusion and Exclusion.

## 9.1 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.

### 9.1.1 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 referencedconceptid from concept_refset AS MSrefset
where MSrefset.refsetId = 32570351000036105
and MSrefset.active = 1
and MSrefset.referencedconceptid in
   (select sourceId from relationships
   where active = 1
   and typeId = 363698007
   and destinationId = 56459004);
```

The result is a much smaller set of less than 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 8) to include all concepts where the finding site is a type of foot structure. That is, replace "destinationId = 56459004" with a subsumption clause like "is_KindOf(destinationId, 56459004)". This query would produce around 800 concepts – all musculoskeletal findings relating to foot structures.

## 9.2 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 using the *Clinical finding grouper exclusion reference set.*

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

### 9.2.1 SQL examples

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

```
select id from concepts
where is_KindOf(id,118185001)
```

This query produces around 1704 concepts[8] – including some that a clinician would not want to record on a patient record. To exclude these, the query can be

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

amended to restrict concepts that are also in the *Clinical finding grouper exclusion reference set*:

```
select id from concepts
where is_KindOf(id, 118185001)
   and id not in (select referencedconceptid from concept_refset
                  where refsetId = 171991000036103);
```

The results from this query now only include 1621 concepts. That is, 83 grouper concepts have been excluded. An example of some of the concepts excluded include:

- 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*|

# 10 Mapping code sets to SNOMED CT-AU

A process for mapping the existing local code sets to SNOMED CT-AU may be required. Careful consideration will need to be given to maintain backwards compatibility with existing data already available in the system. There are two circumstances where mapping may need to be implemented:

1 migration of legacy data to use SNOMED CT-AU terminology; and

2 interoperability use – where HL7 messages are generated from local coded information and translated to SNOMED CT-AU for outbound messaging.

Some considerations are listed below, however NEHTA has produced the *SNOMED CT-AU Mapping Guidelines* [6] which should be read in conjunction with this document.

## 10.1 Mapping options

There are many potential options available to map existing codes to SNOMED CT-AU. The first step to developing or implementing a mapping is to consider what will be required to align the different code sets.

- **One-to-One** – Flat code mapping where there is a simple one-to-one relationship between existing codes and SNOMED CT-AU concepts.

  For example, a local code of “Abrasion-arm” may be mapped to the SNOMED CT-AU concept of “Abrasion of upper limb”.

- **One-to-Many** – When a local code is generic, and needs to be mapped to two or more granular SNOMED CT-AU concepts.

  For example, a local code of “Swelling lips” needs to be mapped to multiple SNOMED CT-AU concepts of “Lip structure”, “Swelling” and “Non Specific”.

- **Many-To-One** – This is when more than one local code can be mapped to a single SNOMED CT-AU concept.

  For example, local codes of “Oromucosal” and “Oral Application” might be mapped to an SNOMED CT-AU concept of “Oromucosal”.

- **No Mapping** – This may require the generation of new content within SNOMED CT-AU; initial support of the local code may not be possible until such new content is generated.

- Any combination of the above.

When mapping to clinical reference sets developed against a NEHTA document specification, the mapping relationship will be dependent on the correlation between the local information model and the specification to which the reference set was developed.

Additionally, prior to commencing any mapping activity it is vital to detail and understand how the data will be used by defining the use cases. This will ensure consistency in the mapping process, creating a reproducible map for the purpose intended.

### 10.1.1 Safety implications

The mapping of terminologies has safety implications and should therefore be considered and conducted in that light. To illustrate:

- When using a specific source term and mapping to a more generic term, care must be taken not to lose clinical significance. For example, if a term of “Diabetes with Renal Failure” at the user interface were mapped to the term “Diabetes” in the messaging system there could be potential patient safety issues.
- Context such as age and sex to ensure the correct diagnosis or treatment plan should be considered. For example, the term “Hyperbilirubinaemia” in an adult patient compared to a neonate (less than 28 days old) has significant impacts on appropriate diagnosis and treatment of a patient.

# 11 Relationship between AMT v3 and SNOMED CT-AU

## 11.1 Modules and dependencies

One of the features introduced with RF2 was modules[9], which enable relatively complex relationships and activities such as:[10]

1 Ownership of components to transfer between organisations, while maintaining the same identifiers;

2 Assembly of custom editions; and

3 Identification of module dependencies.

A SNOMED CT module is defined in the *SNOMED CT Technical Implementation Guide* [1] as:

*A group of SNOMED CT components and/or reference set members that are at a given point in time managed, maintained and distributed as a unit.*[11]

As an example, the core 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 RF2 release.

The two products produced by the AMT v3, SNOMED CT and SNOMED CT-AU are similarly composed of modules as shown in Figure 7.

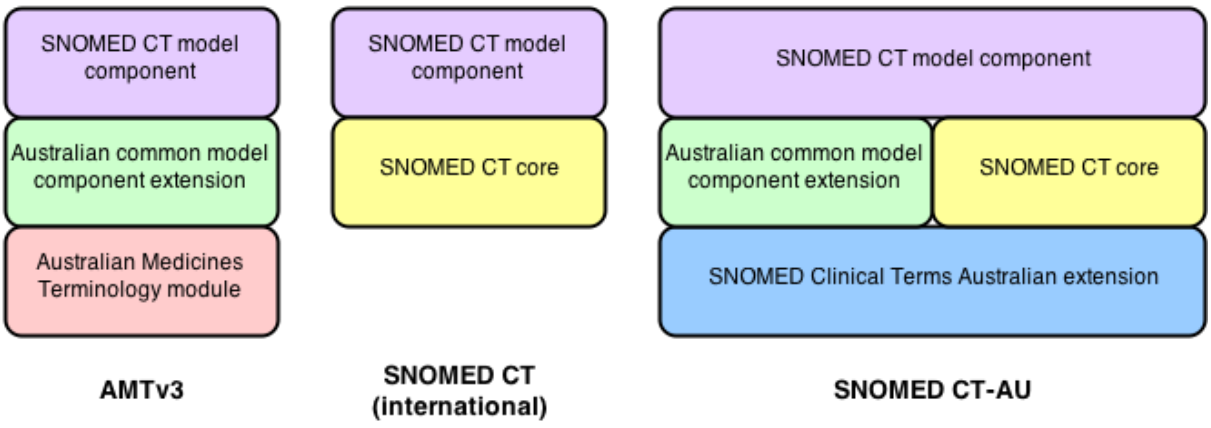


*Figure 7: RF2 modules in terminology releases*

The *Module dependency reference set*[12] specifies the relationship between modules, and in the case of the International Release, shows that the clinical content module is dependent on the metadata module. It is important to note that each release (version) of a module, is dependent on a specific set of versions of other modules.

[9] Section 9.2.1.4.14 "Addition of moduleId field" of the *SNOMED CT Technical Implementation Guide.*
[10] Section 5.2 "SNOMED CT Editions, Extensions, Releases and Modules" of the *SNOMED CT Technical Implementation Guide.*
[11] *IHTSDO Glossary* [7]
[12] Section 7.4.2.4 "Module Dependency Reference Set" of the *SNOMED CT Technical Implementation Guide.*

Consequently the module dependency is updated each release. The November 2014 SNOMED CT-AU release has a dependency chain as illustrated in Figure 8.

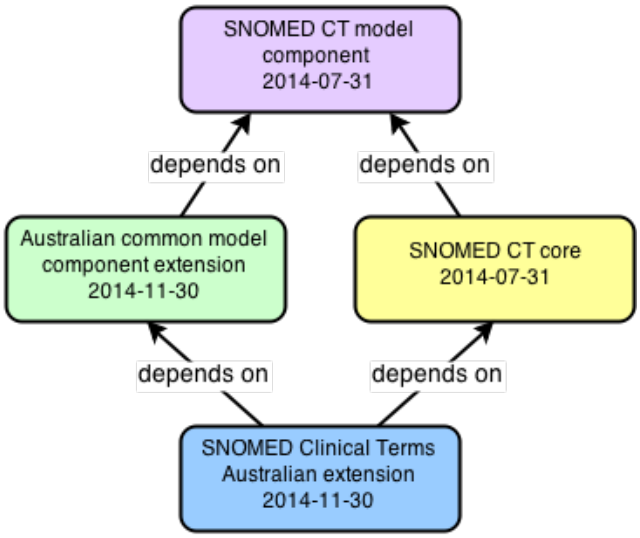


*Figure 8: SNOMED CT-AU module dependency*

## 11.2 Combining SNOMED CT-AU and AMT v3

Section 11.1 shows the modular nature of the NCTIS terminology releases. It can also be seen that there are some modules common to both SNOMED CT-AU and the Australian Medicines Terminology v3 Model (AMT v3). For vendors that need to use both products it is possible to serve both products from a single database, without storing the common (duplicated) modules twice. Essentially, the *Australian Medicines Terminology module* can be appended to SNOMED CT-AU.

Content in the *Australian Medicines Terminology module* can be identified in all the tables as having a value in the *moduleId* field of 900062011000036108.

| id | effectiveTime | active | moduleId | definitionStatusId |
|---|---|---|---|---|
| **446608001** | 20020131 | 1 | 900000000000012004 | 900000000000074008 |
| **32570871000036105** | 20131225 | 1 | 161771000036108 | 900000000000074008 |
| **663001000168106** | 20140831 | 1 | **900062011000036108** | 900000000000073002 |
| **103366001** | 20110131 | 1 | 900000000000012004 | 900000000000074008 |
| **104011000036102** | 20090630 | 1 | **900062011000036108** | 900000000000074008 |

*Figure 9: Example of Concepts file, showing AMT module*

It is also important that if using a Snapshot release of SNOMED CT-AU then the Snapshot release of AMTv3 should also be used, alternatively the full version of each product may be used respectively. The process for assembling a consolidated release will depend on the vendor's implementation approach and available tooling. The content could be extracted from the text files before importing, or ignored on import if a suitable unique key is defined; for example, the *id* field on Snapshot releases.

# Appendix A Transitive Closure Script

The following is a script to create a stored procedure that will generate a transitive closure table, when executed against the Toolkit database referred to throughout this document. The script has been adapted from the one provided in Section 7.7.5.2 “Transitive Closure Implementation” the *SNOMED CT Technical Implementation Guide* [1].

## A.1 Transitive Closure SQL Script

```
/* ------------------------------------------------------------------------
-- Demonstration Transitive Closure creation script
-- Note an alternative (though more verbose) script is available in the
-- SNOMED CT Technical Implementation Guide :
- www.snomed.org/tig?t=tsg2_test_optimizeConcept_transitiveClosure_impl_generate
-- ------------------------------------------------------------------------*/

DELIMITER //

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

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

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

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

    INSERT INTO TransitiveClosure (sourceid,destinationid)
        SELECT DISTINCT b.sourceid,a.destinationid
        FROM TransitiveClosure a
        JOIN TransitiveClosure b ON a.sourceid = b.destinationid
        LEFT JOIN TransitiveClosure 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 idx_TransitiveClosure_sourceid ON TransitiveClosure (sourceid);
CREATE INDEX idx_TransitiveClosure_destinationid ON TransitiveClosure
(destinationid);
```

```
CREATE UNIQUE INDEX idx_TransitiveClosure_primarykey ON TransitiveClosure
(sourceid,destinationid);

END //
```

# References


1. IHTSDO. *SNOMED CT Technical Implementation Guide*. Copenhagen: IHTSDO; 2014. July 2014 release. Available from: http://www.snomed.org/doc.

2. NEHTA. *NCTIS Reference set library*. Sydney: NCTIS; 2014. Release 20141130. Available from: http://www.nehta.gov.au/implementation-resources/ehealth-foundations/snomed-ct-au.

3. NEHTA. *Development approach for reference sets: SNOMED CT-AU*. Sydney: NEHTA; 2014. 20140531 release. Available from: http://www.nehta.gov.au/implementation-resources/ehealth-foundations/snomed-ct-au-common.

4. NEHTA. *SNOMED CT-AU - Reference Set Implementation Toolkit*. Sydney: NEHTA; 2013. v1.0. Available from: http://www.nehta.gov.au/implementation-resources/ehealth-foundations/snomed-ct-au-common.

5. IHTSDO. *SNOMED CT Editorial Guide*. Copenhagen: IHTSDO; 2014. July 2014 release. Available from: http://www.snomed.org/doc.

6. NEHTA. *SNOMED CT-AU - Mapping Guidelines*. 2014. v2.0. Available from: https://www.nehta.gov.au/implementation-resources/ehealth-foundations/snomed-ct-au-common.

7. IHTSDO. *IHTSDO Glossary.* [Internet]. [cited 2014 Aug 22]. Available from: http://www.snomed.org/doc.