SNOMED CT-AU
Australian Implementation Guidance

30 November 2015

Approved for external use

Document ID: NEHTA-2181:2015

National E-Health Transition Authority

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

Acknowledgements

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

IHTSDO (SNOMED CT)

HL7 International

Disclaimer

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

Document control

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

Document information

Key information

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

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. *Reference Set Implementation Guide* integrated.
20141130	30 Nov 2014	Updated for SNOMED CT-AU November 2014 Release.
20150531	31 May 2015	Updated for SNOMED CT-AU May 2015 Release.
20151130	30 Nov 2015	Updated for SNOMED CT-AU Nov 2015 Release. Minor coverage of AMT inclusion.

Table of contents

1 Introduction

1.1 Purpose

This document provides introductory implementation guidance for software developers, technical consumers and the general SNOMED CT®-AU[1] 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* (IHTSDO, 2015) (hereafter: *SNOMED TIG*).

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 reader is assumed to have 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 TIG*. 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* (NEHTA, 2015) and *SNOMED CT-AU Development Approach for Reference Sets* (NEHTA, 2015) describe products that are part of the release whereas this document provides guidance on the options to implement certain aspects of SNOMED CT-AU.

Neither does it cover content already described in the *SNOMED TIG* 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 may not represent the most efficient or robust approach.

Implementers are advised to conduct their own performance tuning and ensure appropriate exception handling.

1.4 Clinical terminology overview

Clinical terminology contributes to the improvement of healthcare through supporting the recording, display and exchange of healthcare information and the ability to deliver decision support services to healthcare providers. Healthcare consumers benefit from the use of terminology to more clearly describe and

[1] "SNOMED" and SNOMED CT" are registered trademarks of the International Health Terminology Standards Development Organisation (IHTSDO).

accurately record their healthcare information. The application of clinical terminology has a range of benefits, including:

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

In addition, clinical terminology supports or enables:

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

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

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

1.5 Related documents

The documents listed below provide the context for development of the reference sets described in this document, and should be read in conjunction with this document to enhance understanding of our approach to terminology development.

- *NCTIS - Development Approach for Reference Sets* (NEHTA, 2015)
- *NCTIS - Reference Set Library* (NEHTA, 2015)

Both documents are available at: http://www.nehta.gov.au/implementation-resources/ehealth-foundations/snomed-ct-au-common.

1.6 Acronyms

Acronym	Description
ADRS	Australian Dialect Reference Set
AMT	Australian Medicines Terminology
DDL	Data Definition Language (also known as "Data Description Language")
IHTSDO	International Health Terminology Standards Development Organisation
NCTIS	National Clinical Terminology and Information Service
RF2	SNOMED CT Release Format 2.0
SNOMED CT-AU	SNOMED CT, Australian release
SQL	Structured Query Language

1.7 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_AustralianImplementationGuidanceSampleScripts.zip** file.

The accompanying sample queries herein use a Snapshot release and filter out active content.

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.

As of November 2015, the Australian Medicines Terminology (AMT) will also be released as part of SNOMED CT-AU in single combined bundle. The AMT will also continue to be published as a standalone product for some time.

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.

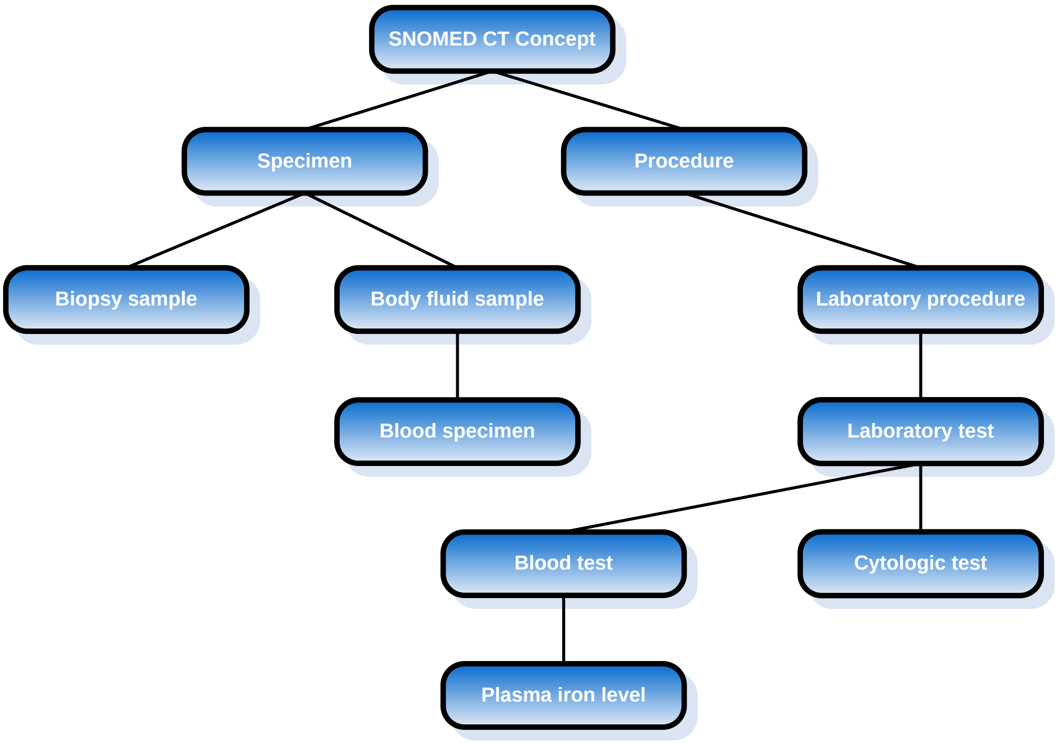


Figure 1: A conceptual view of the SNOMED CT hierarchy

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 TIG.* In addition, Section 3 of this document 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 depicts the four core tables of SNOMED CT that, together with the additional Australian content, comprises SNOMED CT-AU.

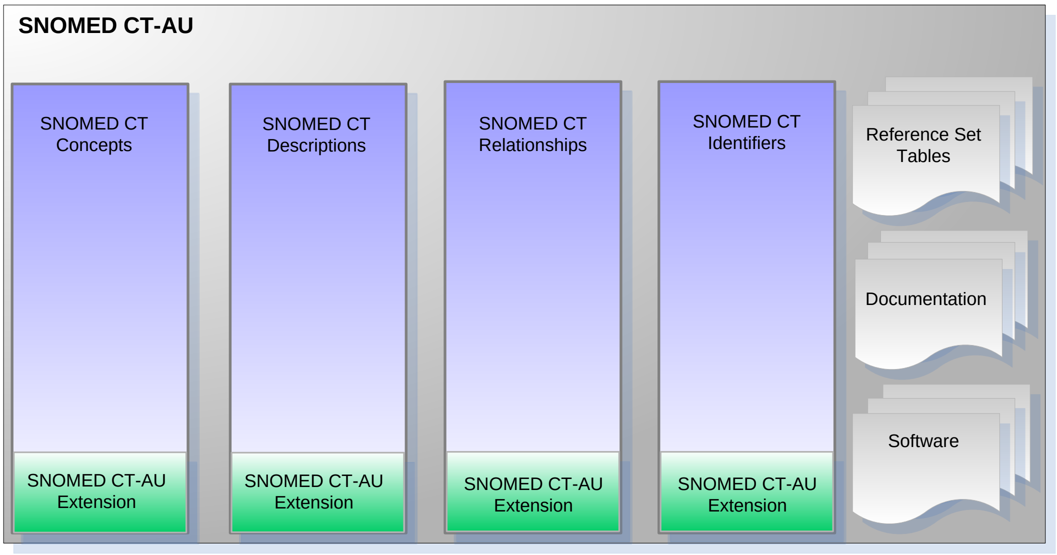


Figure 2: Schematic representation of SNOMED CT-AU

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) is being deprecated and is not supported for use in Australia. For detailed technical specifications and guidance concerning Release Format 2, please consult the *SNOMED TIG.*

Not all of the international reference sets 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.

2.2 Australian Medicines Terminology

The Australian Medicines Terminology (AMT) is the other terminology that is developed and managed by NEHTA. The purpose of the AMT is to provide a consistent and safe approach for the identification and naming of medicines, which can support medicines management and activity in the Australian health domain.

The AMT has been developed for the primary purpose of unambiguously identifying commonly used medicines for clinicians and computer systems in Australia. It can be implemented in clinical information systems to support activities such as prescribing, dispensing and transfer of medications information.

Please refer to the *Australian Medicines Terminology v3 Model Technical Implementation Guide v2.0* (NEHTA, 2014) (hereafter: *AMT TIG*) for further guidance on the implementation of AMT in clinical information systems.

2.3 Relationship between the AMT and SNOMED CT-AU

AMT v3 is a SNOMED CT terminology. It shares the same technical format (RF2) and top level hierarchy as SNOMED CT-AU, and hence the same "semantic space".

2.3.1 Modules and dependencies

One of the features introduced with RF2 was modules[2], which enable relatively complex relationships and activities such as:[3]

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

[2] Section 9.2.1.4.14 "Addition of moduleId field" of the *SNOMED TIG.*

[3] Section 5.2 "SNOMED CT Editions, Extensions, Releases and Modules" of the *SNOMED TIG.*

A SNOMED CT module is defined in the *SNOMED TIG* 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.*[4]

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 RF2 release.

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

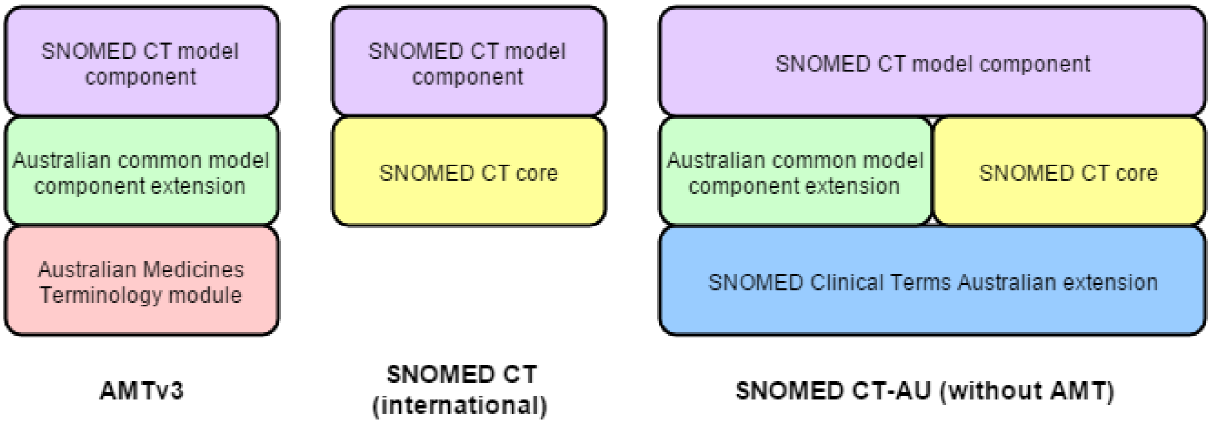

SNOMED CT model component
Australian common model component extension
Australian Medicines Terminology module
AMTv3
SNOMED CT model component
SNOMED CT core
SNOMED CT (international)
SNOMED CT model component
Australian common model component extension
SNOMED CT core
SNOMED Clinical Terms Australian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

Figure 3: RF2 modules in terminology releases

The *Module dependency reference set*[5] 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.

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

[4] *IHTSDO Glossary* (IHTSDO, 2014)

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

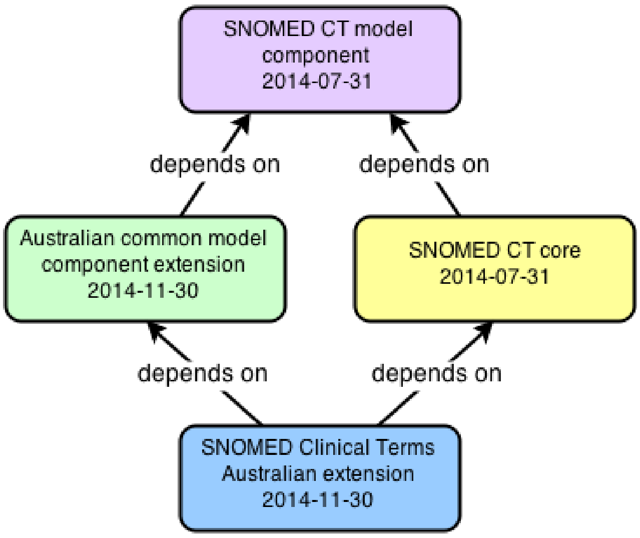

SNOMED CT model component 2014-07-31
depends on
depends on
Australian common model component extension 2014-11-30
SNOMED CT core 2014-07-31
depends on
depends on
SNOMED Clinical Terms Australian extension 2014-11-30

Figure 4: SNOMED CT-AU module dependency (for 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.

2.3.2 Combining SNOMED CT-AU and AMT v3

Figure 3 above shows the modular nature of National Clinical Terminology Information Service (NCTIS) terminology releases. It can also be seen that there are some modules common to both SNOMED CT-AU and AMT v3. As of November 2015, the NCTIS is including the AMT as a part of SNOMED CT-AU. The benefit of this is that vendors who use both products only need to download and manage a single terminology.

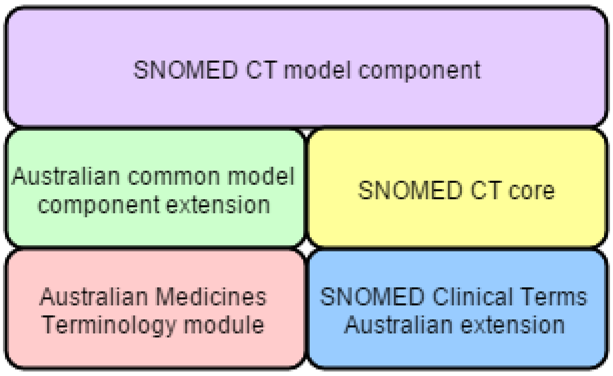

SNOMED CT model component
Australian common model component extension
SNOMED CT core
Australian Medicines Terminology module
SNOMED Clinical Terms Australian extension

Figure 5: Module composition for SNOMED CT-AU combined with the AMT

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

2.3.2.1 Impact of combined release on the length of “term” fields

The length of the “term” fields in SNOMED CT-AU has increased from 256 to 2048 characters to accommodate the combined release of the two terminologies. This larger field size is now required for SNOMED CT-AU too. Otherwise imports of AMT content would fail due to an insufficient number of characters to accommodate the larger term size.

3 SNOMED CT-AU reference sets

3.1 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 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 useable groups. Reference sets can be used for many different purposes, for example:

- Language and dialects can be represented as Language type reference sets. The *Australian dialect reference set* is included in the SNOMED CT-AU release.
- 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:

- Structural reference sets
- Clinical and administrative reference sets
- Bound and non-bound reference sets

3.1.1 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 (for example, the *Australian dialect reference set*);
- the technical metadata associated for the release (for example, module dependencies and reference set descriptors); and
- concrete domains, which provide values for datatype properties.

3.1.2 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.

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

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

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

3.2 Published reference sets

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

3.2.1 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 or administrative reference sets that are not bound to a clinical information specification.

3.2.2 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 will contain more focused content than a Broad context reference set.

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

3.2.3 Specific reference sets

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

3.2.4 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.

SNOMED CT-AU, by way of AMT, includes a unique set of reference sets collectively referred to as Concrete domains reference sets. Such reference sets are used to represent machine-readable values for numbers. In the AMT they are used to represent the magnitude and units associated with medicines, as follows:

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

These reference sets are a crucial part of the concept model and definitions, unlike most other reference sets. As an example, the *Strength reference set* details the strength of a product’s active ingredient with both a value and units (for example, “500 milligrams”) in separate fields. Representing the content as such provides a reliable mechanism for implementers to perform dose calculations and the like.

Please refer to the *AMT TIG* for further details.

3.3 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 where exchange message value domains are involved. 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 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. It may be helpful to refer to the *SNOMED CT-AU Development Approach for Reference Sets*, which contains definitions and development criteria for published reference sets.

- **Maintenance and updates** – 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 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.

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

3.3.1 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.

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 rf2_refset_snapshot AS MSrefset
WHERE MSrefset.refsetId = 32570351000036105
```

```
AND MSrefset.active = 1
AND MSrefset.referencedcomponentid IN
   (SELECT sourceId
   FROM rf2_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 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.

3.3.1.1 Exclusion-based customisation

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

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

SQL examples

Using the example above where a reference set of pregnancy findings is needed, the requirements might be specified as all concepts that are types of 118185001 |*Finding related to pregnancy*| are included. Assuming a subsumption function (as described in Section 8) such as "is_KindOf (candidate,supertype)", such a query may be specified as:

```
SELECT id
FROM rf2_concepts_snapshot
WHERE is_KindOf(id,118185001);
```

This query produces around 1,704 concepts[6] – including some that a clinician would not want to record on 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*:

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

```
SELECT id
FROM rf2_concepts_snapshot
WHERE is_KindOf(id, 118185001)
AND id NOT IN
   (SELECT referencedconceptid
   FROM rf2_refset_snapshot
   WHERE refsetId = 171991000036103);
```

The results from this query now only include 1,621 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*|

3.4 Reference sets used in NEHTA specifications

A commonly described use for reference sets is that which provides a constrained set of concepts to be used within a data field or data element. In NEHTA specifications, where reference sets are stated within value domains, they are intended to state the values which can be recorded for that data element. While these are the intended and encouraged values, the data element's data type defines the actual coding requirements; codes not included in the value domain or even the coding system may still be permitted. Information about representing codes in CDA™[7] is available in *Implementation Guidance - Representing Coding in CDA Documents* (NEHTA, 2011).

Where more than one reference set is listed, the value domain effectively becomes the union of all reference sets and any member of those reference sets may be recorded.

For example, the Shared Health Summary has a *Manifestation* data element with the value domain *Manifestation Values*. The values are all members of the following reference sets:

- 32570071000036102 |*Clinical finding foundation reference set*|
- 142341000036103 |*Clinical manifestation reference set*|

This example shows how multiple reference sets can be combined to gain the desired coverage for a data element and provide a specific value set. The *Clinical finding foundation reference set* provides the broad semantic context and the *Clinical manifestation reference set* provides a smaller set (subset) of the most common manifestation values.

Reference sets that provide a constrained set of concepts to be used within NEHTA specifications have mostly been developed to satisfy particular definitions. This, however, does not preclude their use from other implementations. The definitions upon which they were developed may still be relevant. The *SNOMED CT-AU*

[7] CDA is a trademark of Health Level Seven International and is registered with the United States Patent and Trademark Office.

Development Approach for Reference Sets contains definitions and development criteria for published reference sets and may be helpful when determining a reference set's suitability for use.

3.5 Release cycle

SNOMED CT-AU is 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.

Note: As of November 2015, SNOMED CT-AU content will be released every month, instead of twice yearly, as has been done previously. Each release will contain the AMT content also published that month, as well as additional content specific to SNOMED CT-AU.

3.6 Distribution format and file naming

3.6.1 Distribution format

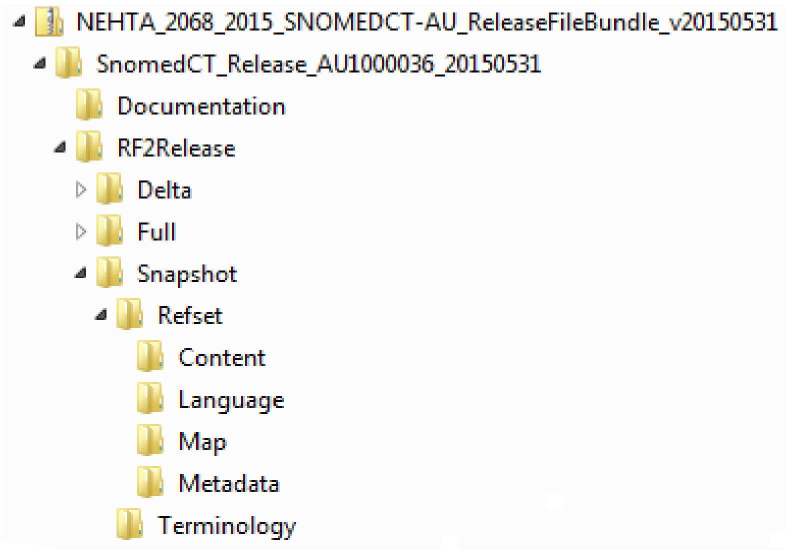
The SNOMED CT-AU release is distributed in a zip file named in the following format:

NEHTA_<identifier>_SNOMEDCT-AU_ReleaseFileBundle_v<release date>.zip

This file contains the following components:

- Documentation
- RF2 Release
  - Delta
  - Full
  - Snapshot

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



```
NEHTA_2068_2015_SNOMEDCT-AU_ReleaseFileBundle_v20150531
  SnomedCT_Release_AU1000036_20150531
    Documentation
    RF2Release
      Delta
      Full
      Snapshot
        Refset
          Content
          Language
          Map
          Metadata
        Terminology
```

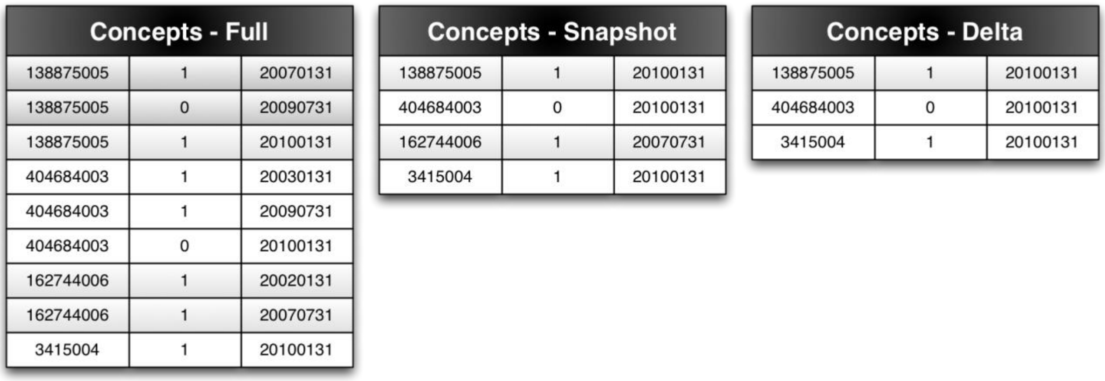
Figure 6: Release bundle folder structure

3.6.2 RF2 distribution types

The RF2 distribution types are comprised of three different forms in a release, as summarised below.

Table 1: Release types

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



Concepts - Full

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

Concepts - Snapshot

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

Concepts - Delta

138875005	1	20100131
404684003	0	20100131
3415004	1	20100131

Figure 7: Example of Full, Snapshot and Delta formats

These forms are useful in different contexts. For example the Snapshot form is easiest to query, however, data updates typically require loading a new Snapshot release. Queries written against the Full form are more complex, however data updates are simplified to appending the next Delta release to the existing Full data.

Implementers need only use one combination of the respective forms, such as a FULL + DELTA or SNAPSHOT + DELTA as per their implementation strategy.

3.6.3 File naming conventions

See the *SNOMED TIG* for further details.

3.7 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 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.

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.

SNOMED CT-AU (including its 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.

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.

However there will 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 0 for further details.

See the *SNOMED TIG* for guidance on importing SNOMED CT into applications.

4.1.2 Reference set implementation considerations

The scope and context within which the reference set is intended to be used needs to be fully understood. This information (much of which is included in this document) is 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 **not** possible to implement reference sets in isolation. The implementation type will determine the extent of terminology data required, but reference sets will always demand the use of additional files. Reasons for this include:

- Referenced concepts have their descriptions held within the Descriptions file. The Descriptions file not only enables the display of terms for concepts, but also the ability to use the wide range of Synonyms required for effective search.
- The *Australian dialect reference set* holds information about the acceptability of Synonyms for use within an Australian context. Most importantly, this language reference set can be used to determine which Synonym is preferred and therefore should be the display term of a reference set member.
- Implementations are required to understand and handle the receipt of unexpected codes. This may occur when the sender or receiver has a more recent release or version of the reference set in their system. If a concept is received, but is not a member of the expected reference set, additional information can be found either in the history of the reference set or in the core terminology files.
- It will assist in the maintenance processes required to keep reference sets up to date (see Section 4.6).

4.2 Data types

Full details of the data types used in the release files are available in the *SNOMED TIG*.

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 (that is, 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 TIG* for related information.

Table 2: 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.

Field	Purpose
refSetId	Uniquely identifies the reference set that this component is a part of. It should be a descendant of *Reference set (Foundation metadata concept)*. This can be found in the *SNOMED CT Model Component (metadata)* hierarchy.
referencedComponentId	Uniquely identifies the component that this row relates to, thus defining membership of this component in the reference set.
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. This is illustrated in the diagram below. Different reference set patterns can be created by adding varying combinations of data type fields to this basic structure. The *SNOMED TIG* details a number of predefined patterns. Over time, NEHTA will release various types (patterns) of reference sets. This document will be extended to describe them as they are introduced.

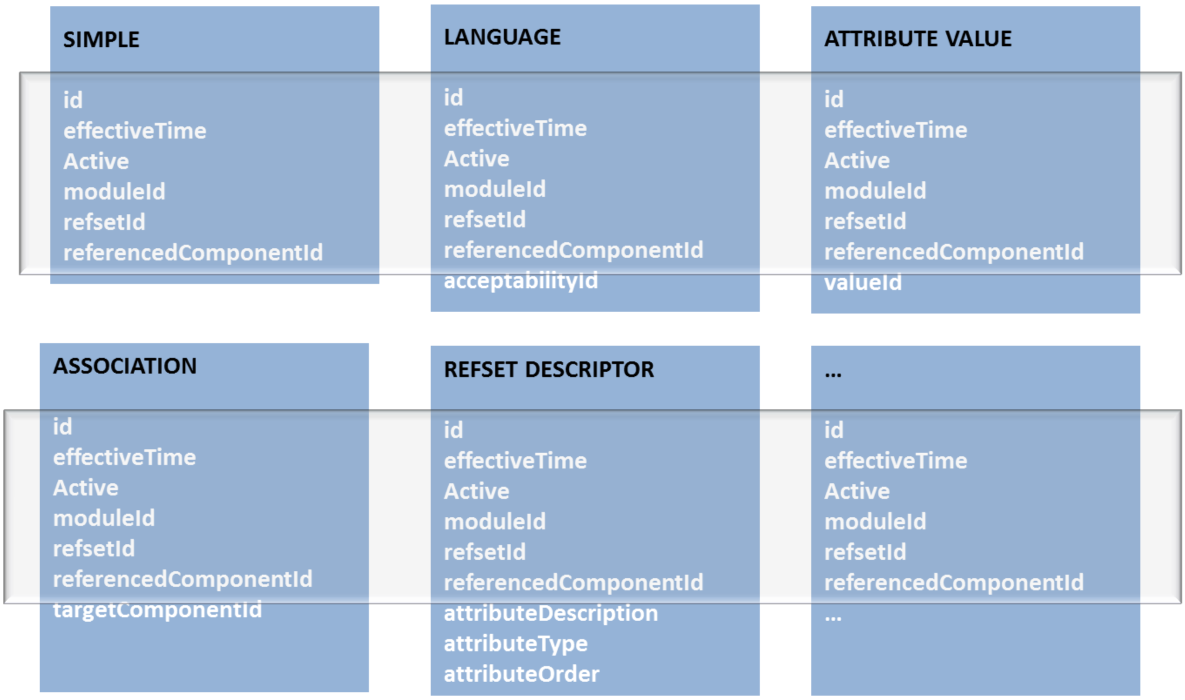


Figure 8: Common attributes of various reference sets

4.4.1 Simple type reference set

This reference set pattern is just the basic reference set member format (see *Table* 2 above) 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 3: 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 expressing the preference and acceptability of descriptions for specific dialects or within 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 preferences 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.[8]

Table 4: Language type reference set format

Field	Purpose
Id	As per Table 2.
effectiveTime	As per Table 2.
Active	As per Table 2.
moduleId	As per Table 2.
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.

Table 5 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* 2 above) are not detailed.

[8] Adapted from Section 5.6.2.8 "Language Reference Set" of the *SNOMED TIG*.

Table 5: 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

As of November 2015, SNOMED CT-AU will be released every month incorporating new content from the international edition and additional nationally authored content, including the AMT. It is the implementer's responsibility to ensure the terminology is current. They should consider release cycles when building import and data transformation processes, or when building applications that use this data. Processes must be adapted to ensure that backwards compatibility between versions can be achieved in their system.

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.

Each SNOMED CT-AU release will include updated core files and updated reference sets. These are identified by SNOMED CT's date-based versioning system. Reference sets are updated to reflect the changes made in the core files. With the exception of the inactivation indicator and historical association reference sets, content reference sets should only have active members which are also active components. In another way, this could be said that, if the latest version of a reference is active, that referenced component should also be active. In addition to ensuring all content is active, it is also important to consider newly created content which may be suitable for use.

As stated, published SNOMED CT-AU reference sets will be updated with every release, however, if using custom SNOMED CT-AU reference sets, this update and maintenance process must be performed by the implementer and preferably in line with release cycles.

4.6.1 SQL examples

As our example, we will use 32570481000036109 |*Emergency department diagnosis reference set*|. In order to identify inactive concepts within the reference set, the following query could be used:

```
QUERY 1

SELECT referencedcomponentid
FROM rf2_refset_snapshot as EDDrefset
INNER JOIN rf2_concepts_snapshot
ON EDDrefset.referencedcomponentid = rf2_concepts_snapshot.id
WHERE EDDrefset.refsetid = 32570481000036109
AND EDDrefset.active = 1
AND rf2_concepts_snapshot.active = 0;
```

The International SNOMED CT January 2014 release inactivated the following concepts, which were referenced by the *Emergency department diagnosis reference set*:

- 105592009 |*Septicaemia*|
- 53869006 |*Gram-negative septicaemia*|
- 301015006 |*Decubitus ulcer of hip*|
- 86216003 |*Mastitis, associated with childbirth*|
- 85292007 |*Postoperative septicaemia*|

Therefore, in the SNOMED CT-AU May 2014 Release, these references were inactivated and suitable replacements were added.

Potential replacements can be determined in two different ways. Concepts could be manually searched for and selected, or a query using the historical association reference sets could be used.

```
SELECT referencedComponentId, refsetId, targetComponentId
FROM association_refset
WHERE referencedconceptid IN (QUERY 1)
AND refsetid IN (900000000000530003,
                 900000000000525002,
                 900000000000524003,
                 900000000000523009,
                 900000000000531004,
                 900000000000526001,
                 900000000000527005,
                 900000000000529008,
                 900000000000528000);
```

Note: Multiple historical association reference sets are listed here. Also the above query is purely illustrative, as the "association_refset" table is not created as part of the sample database setup.

The results are as follows.

Table 6: Concept replacement example

referencedComponentId	refsetId	targetComponentId
105592009	900000000000523009	91302008
53869006	900000000000523009	449082003
301015006	900000000000526001	699214007
86216003	900000000000523009	86196005
85292007	900000000000523009	698819004

Using later examples in Sections 6 and 7, it is also possible to return Descriptions associated with these concept ids. These ids presented as Preferred Terms would look like this:

Table 7: Concept replacement example, with Descriptions

referencedComponentId	refSetId	targetComponentId
Septicaemia	POSSIBLY EQUIVALENT TO association reference set	Sepsis
Gram negative septicaemia	POSSIBLY EQUIVALENT TO association reference set	Sepsis due to Gram negative bacteria
Decubitus ulcer of hip	REPLACED BY association reference set	Pressure ulcer of hip
Mastitis, associated with childbirth	POSSIBLY EQUIVALENT TO association reference set	Disorder of breast associated with childbirth
Postoperative septicaemia	POSSIBLY EQUIVALENT TO association reference set	Postoperative sepsis

These associations aid the identification of potential replacement reference set members. It is important to remember this is just an aid and does not account for all additional new content.

5 Scenarios for implementation

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

5.1 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.

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 (display text) for each concept must also be determined. In most instances, it is strongly recommended that the Preferred Term be used. 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.

5.2 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 is being used in specific messages, these may be published as a "messaging" reference set. To receive a message, all of the messaging reference set data needs to be mapped to the local code sets, which may be a "many-to-one" or "one-to-many" mapping. However, if some SNOMED CT-AU 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 9 for more details on mapping.

5.3 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 TIG* 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.

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 dialect 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 that are useful in a variety of settings.

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

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

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

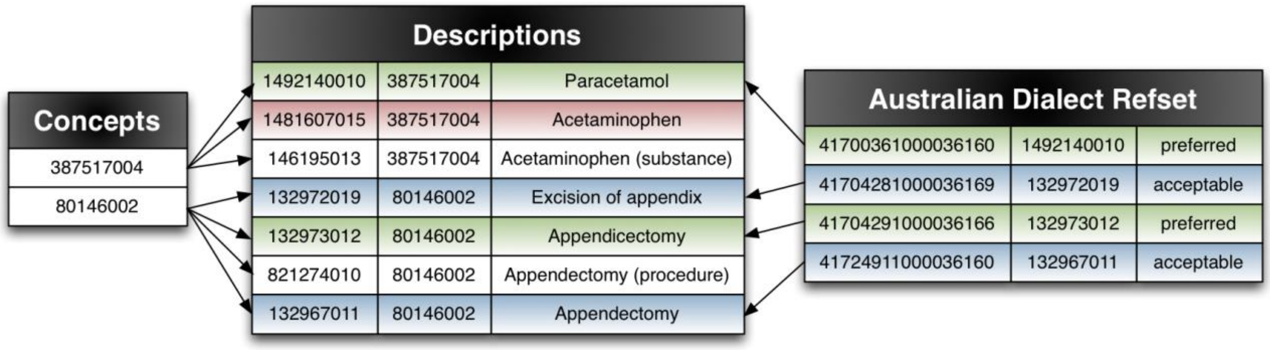
Not Acceptable Indicates that the Synonym is not an acceptable way to describe the concept in this Language type reference set's context. Language type reference sets do not explicitly reference Synonyms with this value, rather "Not Acceptable" is implied by the absence of a reference to those Synonyms.

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 TIG*. As with all reference sets, the referencedComponentId is the component being referenced. For Language type reference sets, this field contains an Id from the Descriptions file. The last field, acceptabilityId, indicates the preference with the current range of values being:

- 900000000000548007 |*Preferred*|
- 900000000000549004 |*Acceptable*|

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



Concepts
387517004
80146002

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

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

Figure 9: Relationship between the Concepts, Descriptions and Language type reference set

Figure 9 above shows an example taken from SNOMED CT-AU, showing how the ADRS expresses that:

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

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

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

6.1.1 SQL examples

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

```
SELECT conceptId,id,term
FROM rf2_descriptions_snapshot
WHERE conceptId = 387517004;
```

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

```
SELECT conceptId,D.id,term, acceptabilityid
FROM rf2_descriptions_snapshot AS D
LEFT JOIN rf2_language_refset_snapshot AS ADRS
ON D.id = ADRS.referencedcomponentid
WHERE D.conceptId = 387517004;
```

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 rf2_descriptions_snapshot AS D
INNER JOIN rf2_language_refset_snapshot AS ADRS
ON D.id = ADRS.referencedcomponentid
WHERE D.conceptId = 387517004
AND ADRS.acceptabilityid = 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 rf2_descriptions_snapshot AS D
INNER JOIN rf2_language_refset_snapshot AS ADRS
ON D.id = ADRS.referencedcomponentid
WHERE D.conceptId = candidate
AND ADRS. acceptabilityid = 900000000000548007
AND ADRS.active = 1);
END
//
```

This function can then be 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 TIG* 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 (approximately 1,800), with acceptable performance (instantaneous):

```
SELECT term
FROM rf2_descriptions_snapshot
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 rf2_descriptions_snapshot
WHERE term LIKE '%frac%';
```

This query will yield approximately 11,000 results that contain "frac" anywhere within the string, however since basic SQL indexing typically relies on the characters at the front of a string – the index is useless and performance will suffer.

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

- **Native full text indexing:** Most relational database management systems (RDBMSs) now provide native full text indexing. The features and syntax vary between systems, so the relevant RDBMS user documentation should be consulted. Such indexes require little extra setup 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[9] is an example of such a library, and versions are available for most major development languages.

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

7.2 Restricting scope to reference sets

An unrestricted search on 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* (IHTSDO, 2015) 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 rf2_descriptions_snapshot \
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 (assuming the reference set has been imported).

```
SELECT term
FROM rf2_descriptions_snapshot
WHERE term LIKE 'ulcer'
AND conceptId IN (SELECT referencedcomponentId
                  FROM rf2_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 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 results 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 "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)*|), 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 "Period pain" would result in unconstrained queries that might look like the following sample.

```
SELECT conceptId, term
FROM rf2_concepts_snapshot AS C
INNER JOIN rf2_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 created function (bolded):

```
SELECT conceptId, get_ADRS_PreferredTerm(conceptId)
FROM rf2_concepts_snapshot AS C
INNER JOIN rf2_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 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 rf2_concepts_snapshot AS C
INNER JOIN rf2_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;[10] including results like 398254007 |*Pre-eclampsia*|, which don't 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.

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

8 Subsumption queries

The concepts in SNOMED CT-AU 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 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 TIG* 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 to match local database schema).

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

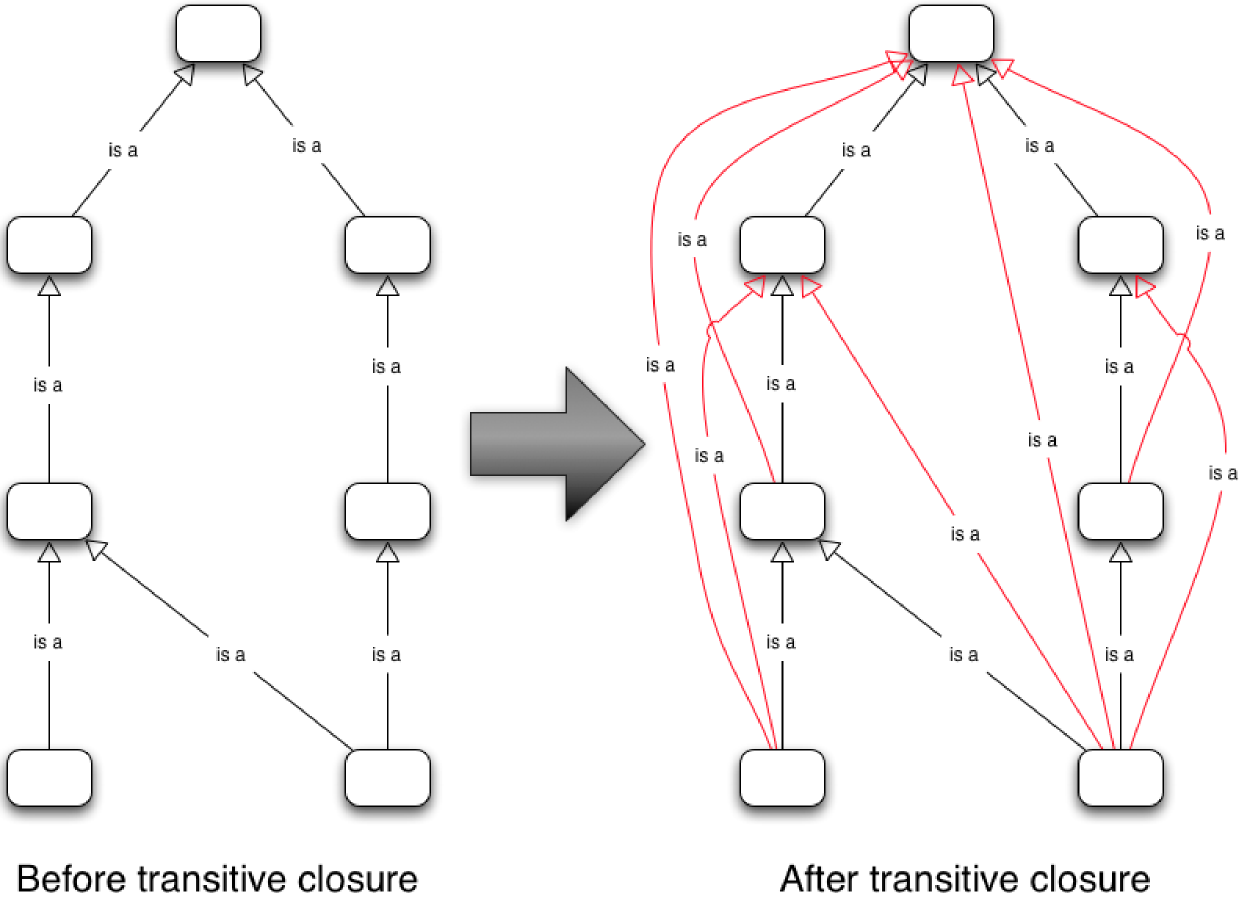


Figure 10: Example of a transitive closure

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 B.

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 rf2_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 8: 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  rf2_transitive_closure
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 9: 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*

sourceId	**destinationId**
Psychogenic dyskinesia	*Perception AND/OR perception disturbance*
Psychogenic dyskinesia	*Motor nervous system finding*
Psychogenic dyskinesia	*Psychological finding*
Psychogenic dyskinesia	*Functional finding*
Psychogenic dyskinesia	*Finding by site*
Psychogenic dyskinesia	*SNOMED CT Concept*
Psychogenic dyskinesia	*Psychological finding of perception*
Psychogenic dyskinesia	*Clinical history and observation findings*
Psychogenic dyskinesia	*Finding of movement*
Psychogenic dyskinesia	*Mental state, behaviour and/or psychosocial function finding*
Psychogenic dyskinesia	*Clinical finding*

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 Section 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 rf2_transitive_closure
WHERE destinationId = 54556006;
```

This query reveals the 45 concepts that are subtypes of this concept including the seven immediate children.[11]

Table 10: 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

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

Immediate Children	All Descendants
	Closed fracture of proximal end of ulna
	Closed fracture of distal end of ulna
	Monteggia's fracture
	Fracture of olecranon
	Fracture of coronoid process of ulna
	Fracture of ulnar styloid
	… and 31more…

8.1.3 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 399907009 |*Animal bite wound*| concepts. Amongst these are 172 409985002 |*Arthropod bite wound*| concepts.

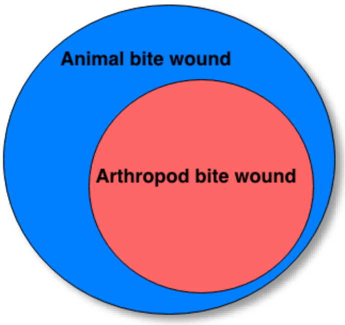


Figure 11: Relationship between Animal and Arthropod bite concepts

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

```
SELECT sourceId
FROM rf2_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 rf2_transitive_closure
WHERE destinationId = 409985002 -- 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[12], 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.

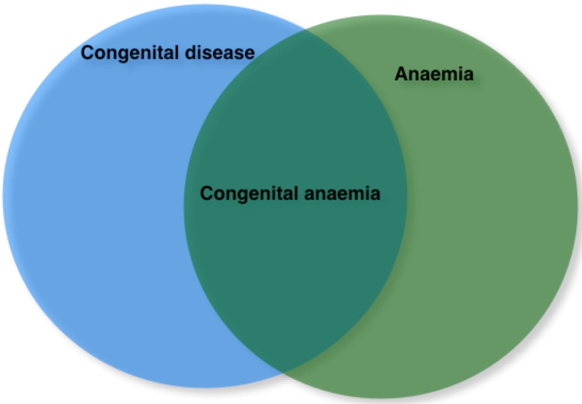


Figure 12: Intersection of Congenital diseases and Anaemias

An inner join query is very well suited to identifying the intersection, as illustrated by the following query:

```
SELECT anaemia.sourceId
FROM rf2_transitive_closure AS anaemia
INNER JOIN rf2_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 8.1.3.

```
SELECT sourceId
FROM rf2_transitive_closure AS anaemia
WHERE anaemia.destinationId = 271737000 -- | anaemia |
AND sourceId IN (
        SELECT sourceId FROM rf2_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.

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

9 Mapping code sets to SNOMED CT-AU

A process for mapping 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; or
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 please consult the *SNOMED CT-AU Mapping Guidelines* (NEHTA, 2014)*,* which should be read in conjunction with this document.

9.1 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 *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 *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 the SNOMED CT-AU concept *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.

9.1.1 Safety implications

Mapping 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 the term “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.
- Contextual information (such as age and sex) should be considered to ensure the correct diagnosis or treatment plan. 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.

Appendix A Sample database schema design

This appendix outlines the content and purpose of the **SNOMED_CT-AU_AustralianImplementationGuidanceSampleScripts.zip** file.

The file contains configuration scripts for the setup of a sample database schema into which the content of the SNOMED CT-AU Release Bundle is loaded. This collection of SQL scripts is intended to provide an educational entry point for loading and working with SNOMED CT-AU in a relational database environment.

Note: These scripts are not a suggested approach for implementing SNOMED CT-AU. Their purpose is to act as a learning tool to understand the release format and data structures of SNOMED CT-AU.

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

Table 11: Configuration scripts

Order of execution	File	Description
1	createReferenceSchema	Contains SQL statements to create the database schema and the core tables.
2	importTables	Contains syntax used to load the SNOMED CT-AU files into the created schema.
3	createIndexes	Contains SQL syntax to create database indexes for the associated tables. These are generally added after importing the data, to maximise the performance of the import.

Note: The files listed in the table above need to be executed in the order in which they are listed.

A.1 Database schema definition

The sample database schema for SNOMED CT-AU content, should consist of a minimum of six tables, namely:

- **Concepts**
- **Relationships**
- **Descriptions**
- **Concept reference set:** This table may contain any number of the Clinical and Foundation reference sets. These entries reference a concept component. If multiple reference sets are imported, they can be filtered using the refsetId field.
- **Description reference set:** This table will contain the *Australian dialect reference set* (ADRS). These entries reference a description component and define a concept attribute value for the membership.

- **Transitive Closure**: This table presents an exploded view of all IS A relationships contained within the terminology.

Note: Identifiers are a pivotal component of the SNOMED CT core and should be included when implementing the database schema in a production environment. However, this table is not required for demonstration purposes.

The following table maps these entities to their respective tables:

Table 12: Entity/table mapping

Entity Name	**Table Name**
Concepts	rf2_concepts_full
Relationships	rf2_relationships_full
Descriptions	rf2_descriptions_full
Concept Reference Set	rf2_refset_full
Description Reference Set	rf2_language_refset_full
Transitive Closure	rf2_transitive_closure

A.2 Entity relationship diagram

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

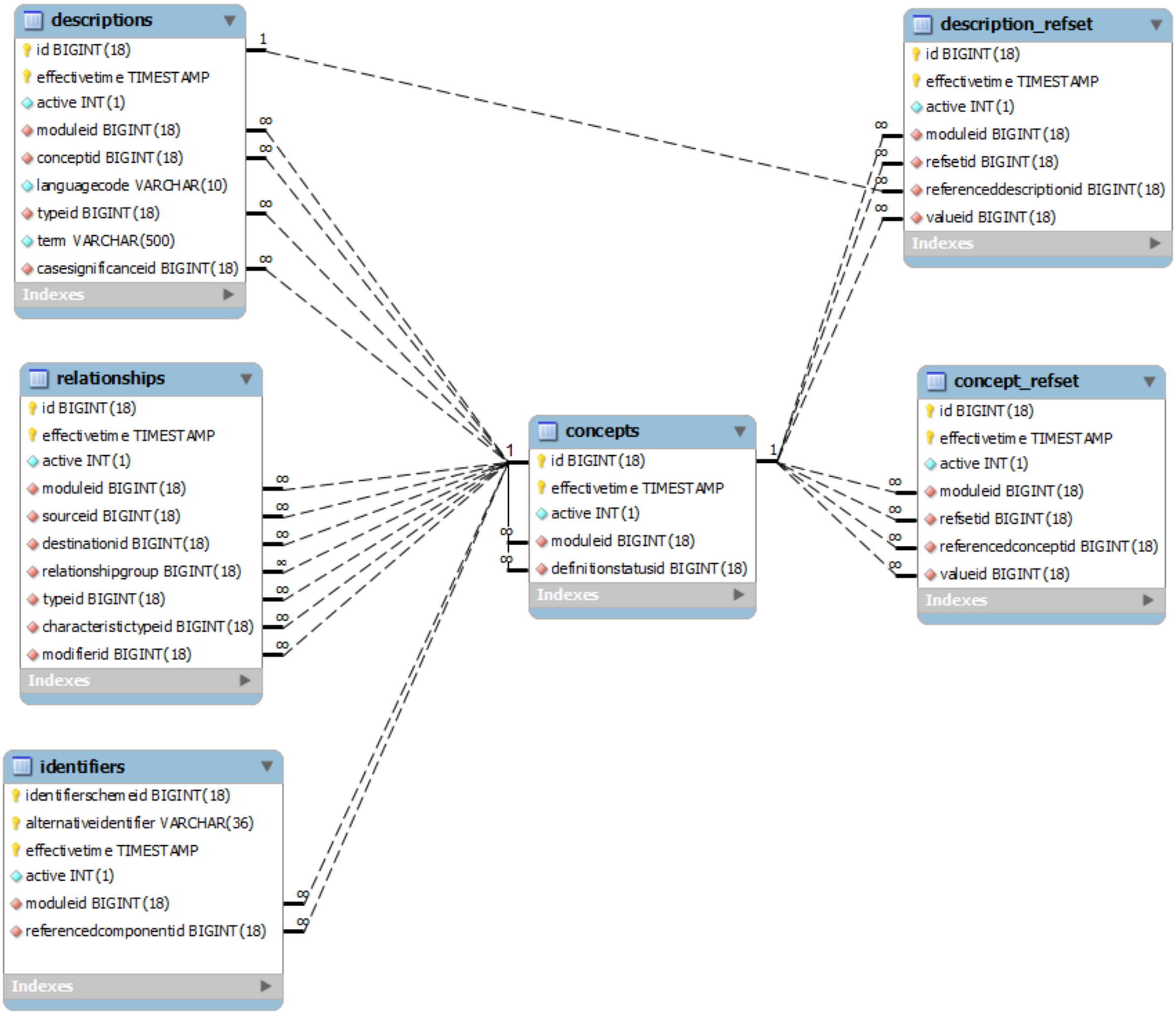


Figure 13: Schema design for core SNOMED CT tables

A conscious decision has been made to create the bare minimum number of tables to allow the sample queries to execute successfully. Hence a custom entity relationship diagram has been created to reflect the simplified set, using the suggested table names used in the creation scripts.

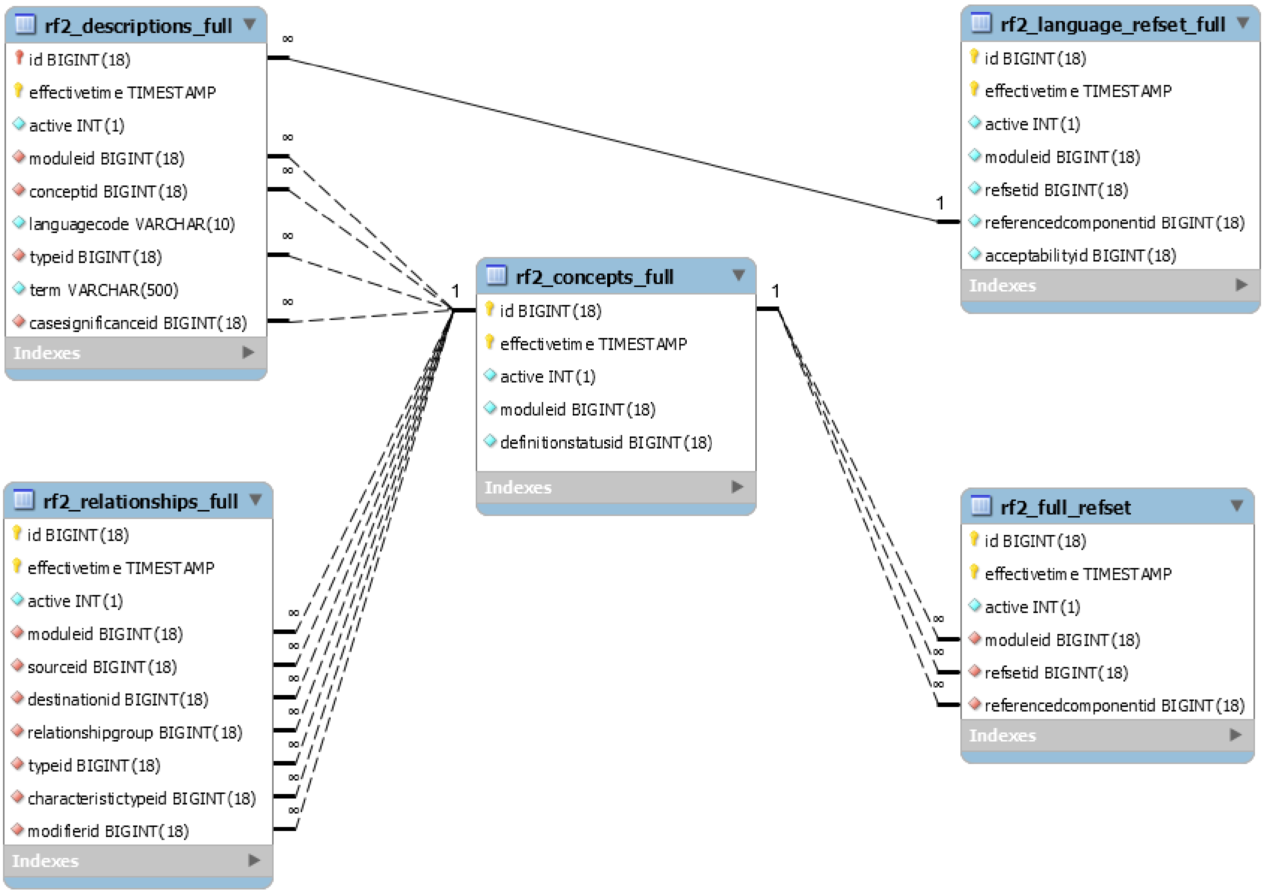


Figure 14: Schema design for SNOMED CT-AU tables

A.3 Database design notes

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

- **MySQL.**

  The table and index creation commands (DDL) and SQL queries[13] contained within the sample scripts file have been constructed within and tested against a MySQL database. This product was selected as a free and easily available choice for a relational database.

- **No referential integrity.**

  Foreign key constraints have been specified in the DDL to define the relationships between tables in the schema. However, at the same time the use of the MyISAM database engine[14] 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.

[13] DDL = Data Definition Language; SQL = Structured Query Language.

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

- **Indexing.**

  Only indexes useful to the SQL queries supplied in the sample code file have been created.

- **No versioning.**

  The sample code file contains scripts required to load both the Full and Snapshot versions of the SNOMED CT-AU release bundle.

Note: The sample queries utilise the Snapshot version of the tables. Accordingly, there is no need to evaluate the effectiveTime column to determine the latest version of a component.

A.4 Notable aspects of the schema creation scripts

A.4.1 FULL Release

The Full version of SNOMED CT-AU content contains the entire history of changes made to each component. A historically complete release of SNOMED CT-AU is loaded into the "rf2_*_full" tables.

A.4.2 SNAPSHOT Release

Consumers of SNOMED CT-AU content would most likely be interested in the most recent version of each released component or the Snapshot release.

The sample scripts file contains two sets of creation and load scripts, namely one each for the Full and Snapshot formats.

All the example queries are to be executed against the Snapshot tables:

- rf2 _concepts_snapshot
- rf2 _descriptions_snapshot
- rf2 _relationships_snapshot
- rf2 _language_refset_snapshot
- rf2_refset_snapshot

Essentially Snapshot tables are derived from the Full tables, using the effectiveTime column to extract the most recent version of each released component.

```
CREATE TABLE rf2_concepts_snapshot AS
SELECT t1.* FROM rf2_concepts_full t1
WHERE t1.effectivetime = (
    SELECT MAX(t2.effectivetime)
    FROM rf2_concepts_full t2
    WHERE t1.id = t2.id);
```

In the statement above, the rf2_concepts_snapshot table is created using a sub-query that extracts the most recent version of every concept from the full release.

A.4.3 DELTA Release

The content of a Delta release should contain entries not in the last Full release. Essentially the Delta files are an update or patch, to allow for the application of any changes since the last release. In effect this would upgrade existing Full or generated Snapshot tables.

Delta release files should be handled as follows:

1 Ensure that the terminology database contains the Full release.

2 Optionally generate a Snapshot set of tables or views from the Full release, using the command syntax listed in the section above. Alternatively an "active" Snapshot view could be created using the following syntax:

```
CREATE OR REPLACE VIEW rf2_concepts_active AS
SELECT t1.* FROM rf2_concepts_full t1
WHERE t1.effectivetime = (
    SELECT max(t2.effectivetime)
    FROM rf2_concepts_full t2
    where t1.id = t2.id)
AND t1.active=1
```

3 Import the Delta entries into the existing Full tables.

4 If Snapshot or active Snapshot tables were created in Step 2, then re-generate them. If views were created instead, they will inherently present the newly added data.

The schema (or DDL) defined in the sample code has been chosen purely to support the queries in the **SNOMED_CT-AU_AustralianImplementationGuidanceSampleScripts.zip** file.

One approach for implementing SNOMED CT-AU could be to create an individual table for each reference set.

A.4.4 Fully Specified Names

Every concept in the SNOMED CT-AU terminology will have one active description of type *Fully specified name (core metadata concept)*, abbreviated as FSN.

Note that FSNs represent the reference point for the meaning of the concept. However, they are not intended to be exposed to users of a clinical system. The following query shows how this term can be extracted for a given concept:

```
SELECT fsn.term
FROM rf2_descriptions_snapshot fsn
WHERE fsn.typeid = (
    SELECT conceptid FROM rf2_descriptions_snapshot
    WHERE term = 'fully specified name (core metadata concept)'
    AND active = 1)
AND fsn.active = 1
AND fsn.conceptid = <id of concept>;
```

A.4.5 Preferred Terms

Every concept in the SNOMED CT-AU terminology will have one or more active descriptions of type *Synonym (core metadata concept)*. Of the Synonyms for each concept, one must be designated as the Australian English preferred description.

This preferred description is the Synonym most suitable for use by end users of clinical systems, although other acceptable Synonyms referenced in the *Australian English language reference set* may also be used.

The preferred Synonym for a given concept is expressed via membership of the *Australian English language reference set*, with the referencedComponentId column being populated with the Description id of the preferred Synonym, and the acceptabilityId column containing the conceptId of the concept *Preferred (foundation metadata concept)*. The following query shows how this reference set can be used to obtain the preferred Synonym for a given concept.

```
SELECT preferred.term
FROM rf2_descriptions_snapshot AS preferred
JOIN rf2_language_refset_snapshot AS member
ON member.referencedcomponentid = preferred.id
AND member.active = 1
WHERE member.acceptabilityid = (
    SELECT conceptid from rf2_descriptions_snapshot
    WHERE term = 'Preferred (foundation metadata concept)'
    AND active = 1)
AND preferred.conceptid = <id of concept>;
```

A.5 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; installation and configuration instructions can be found on the download site.

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

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

SNOMED_CT-AU_AustralianImplementationGuidanceSampleScripts/
         schema/
               Full/
               Snapshot/

         sql/
```

With a current working directory of <some-root-directory>, open a MySQL session.[15] The following statements will create the schema and import the RF2 release files. Also the assumption is that the reader uses the command line version of the MySQL client, rather than MySQL Workbench.

```
mysql> source SNOMED_CT-
AU_AustralianImplementationGuidanceSampleScripts/schema/Full/createReferenceSc
hema.sql

mysql> source SNOMED_CT-
AU_AustralianImplementationGuidanceSampleScripts/schema/Full/importTables.sql

mysql> source SNOMED_CT-
AU_AustralianImplementationGuidanceSampleScripts/schema/Full/createIndexes.sql
```

Please 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. We specifically discourage opening these files in MS Notepad and suggest that you use an editor that honours CR line endings when displaying and writing the file content.
- The `SNOMED_CT-AU_AustralianImplementationGuidanceSampleScripts/schema/Full/importTables.sql` script contains relative paths to the RF2 files. Depending on the operating system and version of MySql, you may need to amend these and replace with the full path. For example:
  `release-files/RF2Release/Full/Terminology/sct2_Concept_Full_AU1000036_20150531.txt`
  changes to
  `C: /Downloads/<SNOMED CT-AU release>/RF2Release/Full/Terminology/sct2_Concept_Full_AU1000036_20150531.txt`

Once the SNOMED CT-AU data has been successfully imported, the content can be retrieved using the sample queries contained within the `SNOMED_CT-AU_AustralianImplementationGuidanceSampleScripts/sql` directory.

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

A.6 Scenarios for Sample SQL queries

The queries discussed in this section are contained in the `SNOMED_CT-AU_AustralianImplementationGuidanceSampleScripts/sql/` directory.

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

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

1 Finding an active concept through a term or description.

```
SELECT *
FROM rf2_descriptions_snapshot
WHERE term LIKE 'Myocardial infarction%'
AND active = 1
ORDER BY conceptid;
```

2 Retrieving the FSN and PT for a particular concept.

```
SELECT
conceptid, term,
CASE typeid
WHEN 900000000000013009
   THEN 'Synonym' ELSE 'Fully specified name' END
   AS description_type
FROM rf2_descriptions_snapshot
WHERE term LIKE 'abdominal aorta finding%'
AND active=1;
```

3 List of preferred descriptions of all active concepts in a particular reference set, such as adverse reaction type.

```
SELECT
c.id AS conceptid,
d.id AS descriptionid,
d.term AS preferred_term
FROM
rf2_concepts_snapshot AS c,
rf2_refset_snapshot AS rs,
rf2_descriptions_snapshot AS d,
rf2_language_refset_snapshot AS adrs
WHERE c.id=rs.referencedComponentId
AND c.id=d.conceptid
AND d.id=adrs.referencedComponentId
AND adrs.acceptabilityid=900000000000548007 -- ID of Preferred Term
AND rs.refsetid= 11000036103 -- ID of Adverse reaction type reference set
AND c.active=1
AND d.active=1
AND rs.active=1
ORDER BY preferred_term;
```

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

```
SELECT
desc_active.term AS "Name of Reference Set",
refset_active.refsetid AS "Reference Set ID",
refset_active.member_count AS "No of Members"
FROM
   (SELECT
   term,id,conceptid
   FROM rf2_descriptions_snapshot AS ds
   WHERE active=1) AS desc_active,

          (SELECT
          referencedComponentId
          FROM rf2_language_refset_snapshot AS lrs
          WHERE refsetId = 32570271000036106 -- Australian dialect refset
          AND acceptabilityid = 900000000000548007  -- Preferred Term
          AND active=1) AS lang_refset_active,

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

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

5 Searching for descendants of a particular concept (106112009 |*Fetal finding (finding)*|) within the terminology.

```
SELECT
c.id AS conceptid,
d.id AS descriptionid,
d.term AS preferred_term
FROM
rf2_concepts_snapshot AS c,

(SELECT sourceId
FROM rf2_transitive_closure
WHERE destinationId=106112009 -- Fetal finding
) AS ffd,

rf2_descriptions_snapshot AS d,
rf2_language_refset_snapshot AS adrs
WHERE c.id=ffd.sourceid
AND c.id=d.conceptid
AND d.id=adrs.referencedComponentId
AND adrs.acceptabilityid=900000000000548007 -- ID of Preferred Term
AND c.active=1
AND d.active=1
AND adrs.active=1;
```

6 Applying 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
rf2_concepts_snapshot AS c,

(SELECT sourceId
FROM rf2_transitive_closure
WHERE destinationId=106112009 -- Fetal finding
AND sourceid NOT IN
   (SELECT referencedcomponentid
   FROM rf2_refset_snapshot
   WHERE refsetid = 171991000036103 -- clinical finding grouper exclusion
refset
   AND active=1
   )) AS ffd,

rf2_descriptions_snapshot AS d,
rf2_language_refset_snapshot AS adrs
WHERE c.id=ffd.sourceid
AND c.id=d.conceptid
AND d.id=adrs.referencedComponentId
AND adrs.acceptabilityid=900000000000548007 -- ID of Preferred Term
AND c.active=1
AND d.active=1
AND adrs.active=1;
```

7 Finding terms within a specific hierarchy.

```
SELECT
d.term AS preferred_term

FROM
rf2_concepts_snapshot AS c,

(SELECT sourceId
FROM rf2_transitive_closure
WHERE destinationId=71388002 -- Procedure hierarchy
) AS pd,

rf2_descriptions_snapshot AS d,
rf2_language_refset_snapshot AS adrs

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

Appendix B Transitive Closure Script

As described in Section 8, 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 type of database (in our case, MySQL), and will likely require modification if used with other providers. The transitive closure can be used for subsumption queries, for example, to find all descendants of 40733004 |*Infectious disease (disorder)*|.

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

B.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 rf2_Transitive_Closure;
    CREATE TABLE rf2_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 rf2_Transitive_Closure (sourceid,destinationid)
        SELECT DISTINCT sourceid,destinationid
        FROM rf2_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 rf2_Transitive_Closure (sourceid,destinationid)
        SELECT DISTINCT b.sourceid,a.destinationid
        FROM rf2_Transitive_Closure a
        JOIN rf2_Transitive_Closure b
      ON a.sourceid = b.destinationid
        LEFT JOIN rf2_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 idx_TransitiveClosure_sourceid ON rf2_Transitive_Closure
(sourceid);
CREATE INDEX idx_TransitiveClosure_destinationid ON
rf2_Transitive_Closure (destinationid);

END //
```

References

IHTSDO. (2014). *IHTSDO Glossary.* Copenhagen: IHTSDO. Retrieved 08 22, 2014, from http://www.snomed.org/doc

IHTSDO. (2015). *SNOMED CT Editorial Guide.* Copenhagen: IHTSDO. Retrieved from http://www.snomed.org/doc

IHTSDO. (2015). *SNOMED CT Technical Implementation Guide.* Copenhagen: IHTSDO. Retrieved from http://www.snomed.org/doc

NEHTA. (2011). *Implementation Guidance - Representing Coding in CDA Documents.* Sydney: NEHTA. Retrieved from https://www.nehta.gov.au/implementation-resources/clinical-documents/common-clinical-document

NEHTA. (2014). *Australian Medicines Terminology v3 Model - Technical Implementation Guide.* Retrieved from http://www.nehta.gov.au/implementation-resources/ehealth-foundations/australian-medicines-terminology-common

NEHTA. (2014). *SNOMED CT-AU - Mapping Guidelines.* Retrieved from https://www.nehta.gov.au/implementation-resources/ehealth-foundations/snomed-ct-au-common

NEHTA. (2015). *NCTIS Development approach for reference sets.* Sydney: NEHTA. Retrieved from http://www.nehta.gov.au/implementation-resources/ehealth-foundations/snomed-ct-au-common

NEHTA. (2015). *NCTIS Reference Set Library.* Sydney: NCTIS. Retrieved from http://www.nehta.gov.au/implementation-resources/ehealth-foundations/snomed-ct-au