SNOMED CT
Release File Specifications

Publication date: 2020-02-06

Web version link: http://snomed.org/rfs

SNOMED CT document library: http://snomed.org/doc

Table of Contents

The SNOMED CT Release File Specification formally defines the formats in which SNOMED CT is provided to licensees (Affiliates). For information about SNOMED CT licensing and availability of release files please refer to the Member Licensing and Distribution Service(MLDS) (http://snomed.org/licensing).

Web browsable version: http://snomed.org/rfs

SNOMED CT Document Library: http://snomed.org/doc

1. Introduction

Background

SNOMED CT is the most comprehensive and precise clinical health terminology product in the world. It has been developed collaboratively to ensure it meets the diverse needs and expectations of clinicians worldwide and is accepted as a common global language for health terms. SNOMED CT enables meaning-based retrieval and processing of clinical data because it applies description logic techniques to represent the meaning of clinical concepts. As a result, patients and healthcare professionals benefit from improved health records, clinical decisions and analysis, leading to higher quality, consistency and safety in healthcare delivery.

SNOMED International distributes SNOMED CT to its licensees as a package of release files that can be loaded into software applications that enable and optimize access to the terminology.

Purpose

This document provides the formal specification of the file structures in which SNOMED CT is distributed. The specified file structures apply to the releases of the SNOMED CT International Edition, and to SNOMED CT extensions distributed by Members and Affiliates .

Audience

The intended audiences for this guide are those involved in the creating or consuming SNOMED CT release files. This includes:

- SNOMED International staff involved in creating or quality assuring SNOMED CT release files.
- National Release Centers and Affiliates that create, maintain and distribute SNOMED CT extensions.
- Software designers and developers responsible for SNOMED CT enabled software applications that need to access data provided in SNOMED CT release files.
- Anyone interested in a detailed understanding of the logical design of SNOMED CT and way that logical design is represented in release files.

Notes

Licensing Note
This guide refers to files that are included in the International Release of SNOMED CT provided to licensees by SNOMED International. It also refers to additional files that are included in SNOMED CT extensions provided by Members and Affiliates..
Details of the licensing conditions for SNOMED CT are available from the SNOMED International web site (www.snomed.org).

Update Note
Starting on 31 July 2018, two new reference sets are being introduced to provide a more expressive representation of concept definitions. Initially these files will only contain supplementary information that cannot be represented in the current stated relationship file. From 2019 onward, these new references sets will include the full representation of the stated view of all concept definitions and on completion of the transitional process the stated relationship file will be deprecated.
Further details of this change are provided in the relevant sections of this specification. As this version to the guide is being published at the start of the transitional period, it includes specifications of current files and newly introduced files.

Historical Note

This document specifies release file structures known as Release Format 2 (RF2). These file structures were introduced in January 2012 to support built-in version tracking and more flexible extensibility mechanisms, including reference sets and packaging of release files into separately identifiable modules. Between its initial release in January 2002 and January 2012, SNOMED CT release files were structured in accordance with an earlier file structure specifications that are now referred to a Release Format 1 (RF1). That release format is deprecated not longer used or supported by SNOMED International.

2 SNOMED CT Logical Model

This section outlines the logical model at the heart of the design of SNOMED CT. It then provides a summary of the ways in which different elements of this logical model are represented in release files. Finally, it provides an overview of the concept definitions that provide the semantic foundation for meaningful processing of clinical information.

2.1 High Level Logical Model of SNOMED CT

Figure 2.1-1 provides a high-level view that illustrates the concept-centric design of SNOMED CT. The subsections below describe the different elements illustrated here.

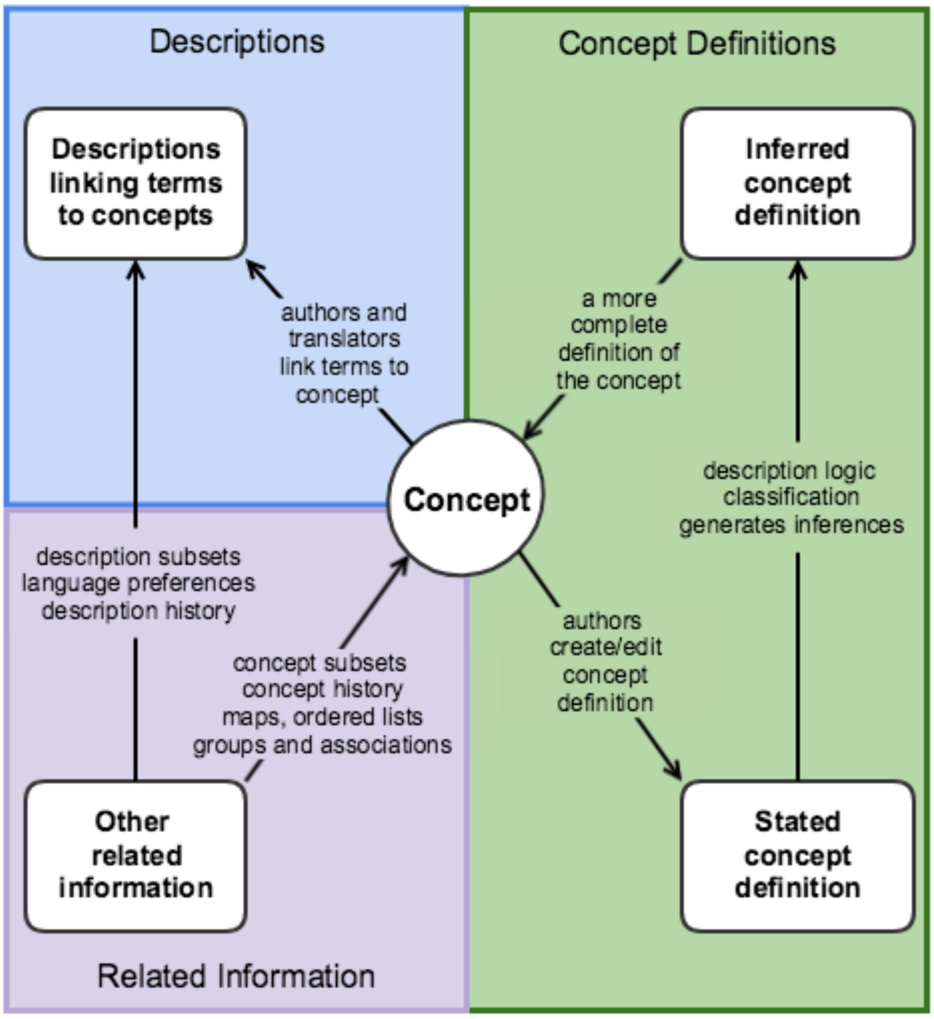

Descriptions
Concept Definitions
Descriptions linking terms to concepts
Inferred concept definition
authors and translators link terms to concept
a more complete definition of the concept
Concept
description logic classification generates inferences
description subsets language preferences description history
concept subsets concept history maps, ordered lists groups and associations
authors create/edit concept definition
Other related information
Stated concept definition
Related Information

Figure 2.1-1: High-level abstract view of the design of SNOMED CT

Concepts

A concept is defined as a clinical idea to which a unique concept identifier has been assigned.

Notes

- *SNOMED CT concepts* are distributed in the concept file.
- *Concepts* are associated with descriptions that contain human-readable terms describing the concept.
- *Concepts* are related to one another by relationships and OWL axioms that provide a formal logical definition of the *concept*.

Authors create a new SNOMED CT concept for each distinct clinical meaning added to the terminology.
Every concept is uniquely identified and this identifier allows the concept to be unambiguously recorded in a clinical record. The concept identifier also allow other related information to be linked to a specific concept. This linked information includes human-readable terms and formal concept definitions.

Descriptions

A description is defined as an association between a human-readable phrase (term) and a particular SNOMED CT concept.

Notes

- Each *description* is represented by a separate row in the

  Error rendering macro 'sp-plaintextbody-link'

  Conversion context did not contain original content entity.

  .
- Each *description* has a unique identifier and connects a concept with a *term* of a specified description type. All concepts have descriptions with description types fully specified name and synonym. Other description type can be defined and may be applied to some concepts.

Terminology authors create a set of descriptions each of which links a term to an identified concept. The linked terms must all be legitimate ways to label or refer to the concept to which they are linked. Translators also create descriptions to link appropriate terms in other languages or dialects to the same concepts. Since usage of terms varies depending on languages and dialects, the design also support addition of information about which descriptions contain terms that are preferred or acceptable in each language or dialect.

Concept Definitions

Terminology authors associate each concept with a formal stated definition. This stated concept definition consists of description logic axioms that are known to be true for that concept (for example stating that appendectomy is a procedure that applies the method excision to the appendix structure). Because these axioms are stated in a formal way, a description logic classifier can be applied to all the axioms in the terminology to generate additional logical inferences. Therefore, the overall model is designed to enable representation of both the original stated concept definition and the inferred view of the concept definition.

In practice, a description logic classifier can generate more than one inferred view, depending on whether all logically inferred axioms are retained and on whether some types of redundant axioms are omitted from the inferred view.

Other Related Information

The SNOMED CT design also enables a wide range of customizable information to be linked to the concepts and descriptions mentioned above. The extensible design of SNOMED CT enables consistent representation and distribution of:

- subsets of concepts or descriptions
- ordered lists of concepts or descriptions
- language preferences associated with different descriptions
- ordered or unordered associations between concepts
- coded or plain text annotations linked to specific concepts
- maps from concepts to codes in other code systems
- representation of queries and expressions

2.2 Representation of the Logical Model

Figure 2.2-1 shows how SNOMED CT release files represent the logical model following completion of a migration period between July 2018 and July 2019. Note that the only structural change was the replacement of the stated relationship file with the OWL expression reference set files. The representation of the inferred view of concept definitions is unchanged from the perspective of the release file structure. However, the nature and quality of the inferred relationships will changes as a result of inferences derived from the enhanced definitions represented as axioms in the OWL axiom reference set.

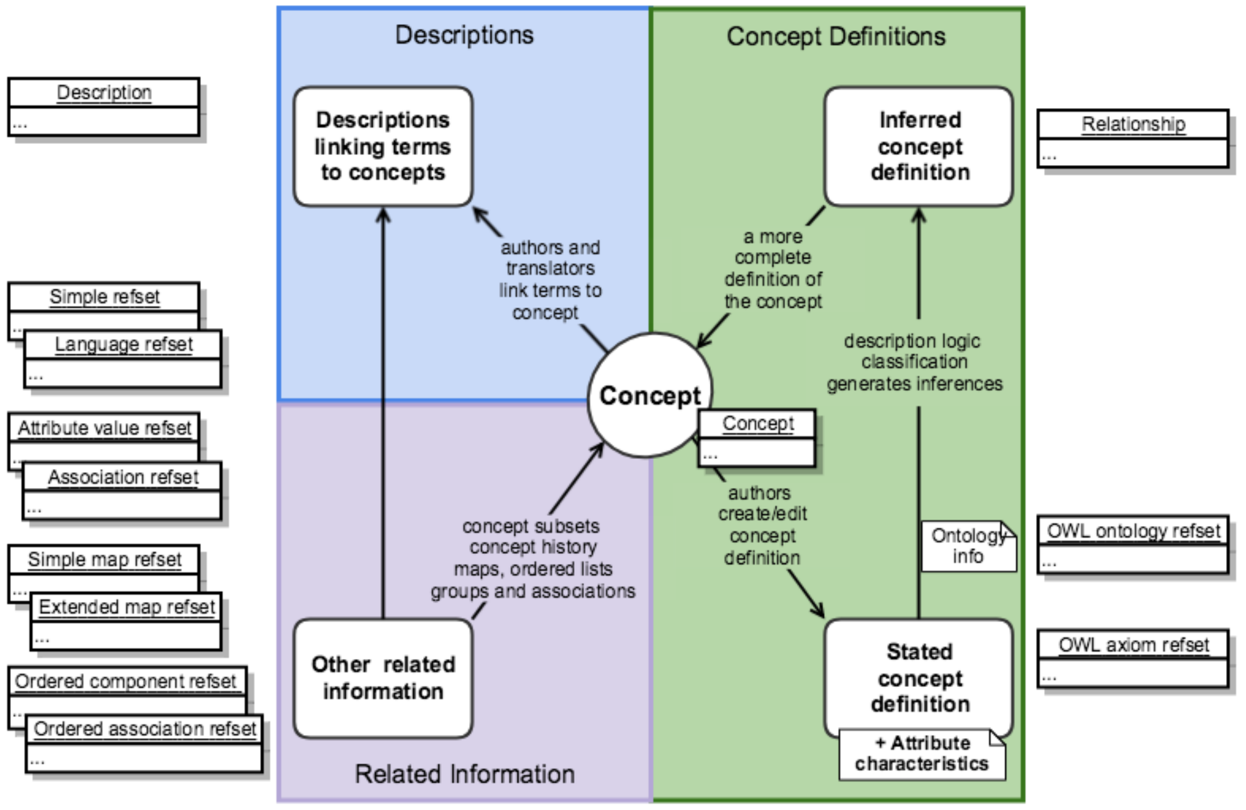

Descriptions
Concept Definitions
Description
...
Descriptions linking terms to concepts
Inferred concept definition
Relationship
...
authors and translators link terms to concept
a more complete definition of the concept
description logic classification generates inferences
Concept
Concept
...
Simple refset
Language refset
Attribute value refset
Association refset
Simple map refset
Extended map refset
Ordered component refset
Ordered association refset
concept subsets concept history maps, ordered lists groups and associations
authors create/edit concept definition
Ontology info
OWL ontology refset
OWL axiom refset
Other related information
Stated concept definition
+ Attribute characteristics
Related Information

Figure 2.2-1: Representation of the logical model of SNOMED CT

Table 2.2-1: Release file representation of the logical model

Logical Model	Release File Representations	References
Concepts	Each concept is represented by a row in the concept release file.	4.2.1 Concept File Specification
Descriptions	Each description is represented by a row in the description release file.	4.2.2 Description File Specification

Stated Concept Definitions	Each stated concept definition is represented by a set of rows in the OWL axiom reference set file, which follows the format of an OWL Expression Reference Set. Each row contains an axiom that forms part of the definition of the concept identified by the referencedComponentId. **Notes:** • As well as representing the definitions of individual concepts, the OWL axiom reference set represents characteristics of attributes including transitivity, reflexivity and property chains. • The OWL ontology reference set also follows the OWL Expression Reference Set pattern. It contains general information about the terminology, which is required by a description logic classifier but is not subject to significant changes between release versions. **Change Note** This representation was introduced in July 2018 and, following a transitional period, now fully represents all stated concept definitions.	5.2.21 OWL Expression Reference Set SNOMED CT OWL Guide SNOMED CT Logic Profile Specification
Inferred Concept Definitions	Each inferred concept definition is represented by a set of rows in the relationship release file. Each row in the set that defines a concept, represents a necessary, defining relationship with another concept. The definitionStatusId column in the concept file row indicates whether the set of defining relationships is sufficient to define the concept.	4.2.3 Relationship File Specification
Other Related Information	Represented by a range of reference set release files that conform to the extensible reference set file format. Each row in a reference set refers to a concept or description as a member of the set.The extensible structure allows different types of related information to be associated with the referenced component.	5.2 Reference Set Types Practical Guide to Reference Sets

More Information About Concept Definition Updates

In this document

- 5.2.21 OWL Expression Reference Set

In other documents

- SNOMED CT OWL Guide
- SNOMED CT Logic Profile Specification

In E-Learning Presentations

- Updates to Support Advanced Description Logic

Historical Note on Logical Model Prior to July 2018

- Representation of the Logical Model - Before July 2018

2.3 Concept Definitions

Prior to the 31 July 2018 release, documentation about concept definitions focused on the central role of defining relationships. Updates to enable the use of more advanced description logic features mean that some aspects of concept definitions cannot be fully represented by defining relationships. Therefore, this section introduces and adopts a new focus on concept definitions and the assertions (or axioms) that form the building blocks of these definitions.

- On completion of the updates, the stated relationship file will be deprecated. From that point forward the OWL axiom reference set file will be the standard distribution file for the stated view of concept definitions.
- The relationship file will continue to be used to distribute the inferred view of concept definitions.

Glossary Definition
A concept definition is A set of one or more <a href="https://confluence.ihtsdotools.org/display/DOCGLOSS/axiom" title="Glossary link: axioms" rel="nofollow" class="conf-macro output-inline" data-hasbody="false" data-macro-name="gloss">axioms</a> that partially or sufficiently specify the meaning of a <a href="https://confluence.ihtsdotools.org/display/DOCGLOSS/SNOMED+CT+concept" title="Glossary link: SNOMED CT concept" rel="nofollow" class="conf-macro output-inline" data-hasbody="false" data-macro-name="gloss">SNOMED CT concept</a>.</p> <div class="NoPrint" style="margin-top:20px;"> a set of one or more axioms that partially or sufficiently specify the meaning of a SNOMED CT concept.

Notes

- The axioms that specify a *concept definition* are represented in release files as SNOMED CT relationships or as OWL axioms that conform to the OWL Functional Syntax.

This section outlines some of the features of concept definitions and the impact of different ways of representing concept definitions on the precision and completeness of those definitions. This section has been kept as brief as possible and is only intended to raise awareness of the changes. For those interested in understanding more about each of the topics mentioned, a more detailed appendix is provided which illustrates each of the points made in this section.

Overview of this Section	Overview of Supporting Appendix
	Appendix D: Concept Definition Illustrations

2.3.1 Stated and Inferred Concept Definitions

This section briefly outlines two distinct types of views of SNOMED CT concept definitions. More detailed illustrations of some of the points on this page are provided in D.1 Stated and Inferred Definitions - Examples.

Stated View of Concept Definitions

SNOMED CT concepts are defined by assertions made by SNOMED CT authors. The concept definitions asserted by SNOMED CT authors are known as the stated view.

The stated view is a representation of concept definitions consisting only of assertions made or revised by SNOMED CT authors.

Notes

- In contrast to the inferred view, the *stated view* does not include inferences generated by applying a description logic classifier.

Description Logic Classification

A description logic classifier can apply logical rules to the stated view to create inferences. The end result of this process is an inferred view of concept definitions.

Inferred Views of Concept Definitions

The inferred view is a representation of concept definitions that is logically derived by applying a description logic classifier to the stated view.

Notes

- Different *inferred views* can be derived from the same stated view by applying different rules that selectively exclude some types of assertions.
- Different *inferred views* may be semantically equivalent to one another provided that assertions are only excluded if they are redundant (i.e. can be *inferred* from assertions that are included). However, in some cases, an *inferred view* may not completely represent the concept definition but may serve a specific purpose.

Choosing the View to Use

The decision on whether you need access to the stated view and/or inferred view depends on your answers to the following questions.

If you answer "yes" to any of the following questions you need access to the stated view

1. Do you already use or plan to use a description logic classifier with SNOMED CT?
2. Are you responsible for or planning to create a SNOMED CT extension in which you will add new clinical concepts?
3. Do you need to query or analyze postcoordinated expressions in records?
4. Are you interested in experimenting with the application of description logic to SNOMED CT (e.g for educational reasons)?

If you answered yes to one these questions above, you should still check the next set of questions to consider if you would also benefit from accessing the inferred view. On the other hand, if you apply a classifier to the stated view you could generate your own inferred view.

If you answer "yes" to any of the following questions you need access to an inferred view

1. Do you need access to a simple relational table representation of the defining relationships between concepts?
2. Do you need to be able to display or navigate the subtype hierarchy using only the links between each concept and its proximal subtypes and proximal supertypes?
3. Do you need to generate a transitive closure view of the subtype hierarchy to assist with rapid subsumption testing?

Several different inferred views can be derived from a single stated view. The inferred view distributed as part of the SNOMED CT release conforms to the necessary normal form[1].

If you answer "yes" to any of the following questions you do not need direct assess to either view.

1. Does an application you are responsible for use software or services provided by another organization to access SNOMED CT?
2. Are you an end-user of a software application that provides access to SNOMED CT?

---

[1] The necessary normal form (NNF) is similar to the distribution normal form (DNF) that was released prior to the updates in 2018. However, the rules for Generating Necessary Normal Form take account of the need to omit more advanced features that cannot be represented as relationships.

2.3.2 Necessary Conditions and Sufficient Definitions

This section briefly outlines different aspects of assertions made about concepts and the extent to which they are necessarily true or form part of a sufficient definition of the concept. More detailed illustrations of some of the points on this page are provided in D.2 Necessary and Sufficient - Examples.

Assertions

The stated view of concept definition consists of one or more assertions made by SNOMED CT authors.

Necessary Conditions

Each time an assertion is made about a concept, an author must decide if that assertion is a necessary condition. If the assertion is always true for that concept and its subtypes, it is a necessary condition.

- This implies that for all instances of that concept or its subtypes, the assertion must be true, even if it has not been explicitly stated.

A necessary condition is defined as a characteristic that is always true of a concept.

Example

- If you have a 71620000 |fracture of femur|, the morphological abnormality 72704001 |fracture| must be present. Therefore, 116676008 |morphology| = 72704001 |fracture| is a *necessary condition* of 71620000 | fracture of femur|.

Sufficient Definitions

For each concept an author must decide if there are one or more sets of assertions that form a sufficient definition of that concept. A set of assertions is a sufficient definition if it distinguishes a concept and its subtypes from other concepts.

- This implies that if all assertions in the set are true for a concept, it must be an instance of the defined concept or a subtype of that concept.

A sufficient definition is a set of characteristics which distinguish a concept and its subtypes from all other concepts.

Notes

- Any concept that matches the *sufficient definition* is equivalent to or a subtype of the defined concept.
- A concept may have more than one *sufficient definition*. In that case any concept that matches at least one of these *sufficient definitions* is equivalent to or a subtype of the defined concept.

Examples

- The following set of assertions is a sufficient definition for 74400008 |appendicitis (disorder)| because any concept for which this set of assertions is true must either be the disorder *appendicitis* or a subtype of *appendicitis*.

```
18526009 |disorder of appendix| +
    302168000 |inflammation of large intestine| :
    116676008 |associated morphology| = 23583003 |inflammation| ,
    363698007 |finding site| = 66754008 |appendix structure|
```

- Both the following sets of assertions are sufficient definitions for the concept 8801005 |Secondary diabetes mellitus (disorder)|:

```
73211009 |Diabetes mellitus| : 246075003 |Causative agent| = 105590001 |Substance|
```

```
73211009 |Diabetes mellitus| : 42752001 |Due to| = 64572001 |Disease|
```

- While each of the assertions 246075003 |Causative agent| = 105590001 |Substance| and 42752001 |Due to| = 64572001 |Disease| form part of a sufficient definition, neither of these assertions are necessary conditions because *only one* of them needs to be true. This illustrates that an assertion that is part of a sufficient definition need not be a necessary condition.

Concepts with no Sufficient Definitions

A concept that has no sufficient definitions is a primitive concept.

Because primitive concepts have no sufficient definitions it is not possible for a description logic classifier to determine if other concepts are subtypes of this concept. Similarly, it is not possible to automatically determine whether an expression is a subtype of a primitive concept. Therefore, only concepts or expressions that explicitly state they are subtypes of primitive concepts will be treated as subtypes when applying expression constraints or undertaking analysis.

However, note that this does not prevent a primitive concept being classified as a subtype of a sufficiently defined concept.

Concepts with a Sufficient Definition

A concept that has at least one sufficient definition is a sufficiently defined concept.

A description logic classifier can determine whether the stated definitions of other concepts meet at least one of the sufficient definitions and if so will classify these concepts as its subtypes. Similarly, it is possible to determine whether an expression is equivalent to or a subtype of a sufficiently defined concept. Therefore, where expression constraints or queries refer to sufficiently defined concepts the results will include the inferred subtypes of these concepts.

Sufficiently Defined Concepts with Necessary Conditions

If a sufficiently defined concept has one or more additional necessary conditions then any concept or expression that satisfies one of its sufficient definitions will also inherit any necessary conditions.

For example one sufficient definition of 397825006 |Gastric ulcer (disorder)| is an ulcer in a stomach structure:

```
=== 64572001 |disease| :{ 116676008 |associated morphology| = 56208002 |ulcer| ,
    363698007 |finding site| = 69695003 |stomach structure| }
```

However, another definition could be created with a more specific site gastric mucosa:

```
=== 64572001 |disease| :{ 116676008 |associated morphology| = 56208002 |ulcer| ,
    363698007 |finding site| = 78653002 |gastric mucosa| }
```

In both cases these definition are equivalent to 397825006 |Gastric ulcer (disorder)|. The more general definition is flexible when it comes to allowing refinement to a specific location of the ulcer within the stomach, which is actually useful information. It also avoids requiring an expression to refer specifically to the mucosa (stomach lining), which is where all gastric ulcers occur.

For example, an expression including the specific location could look like this

```
=== 64572001 |disease| :{ 116676008 |associated morphology| = 56208002 |ulcer| ,
    363698007 |finding site| = 127869006 |Anterior wall of fundus of stomach| }
```

This satisfies the sufficient definition because the finding site is a subtype of stomach structure. This will therefore classify as a type of 397825006 |Gastric ulcer (disorder)| located in the anterior wall of the gastric fundus. The problem is that a query for disorders of the gastric mucosa will not find this expression.

<< 64572001 |disease| : 363698007 |finding site| = 78653002 |gastric mucosa|
However, adding the definition that refers to the gastric mucosa as an additional necessary condition can solve this problem. The expression satisfies the sufficient definition implying this is a type of 397825006 |Gastric ulcer (disorder)|. The fact that it is a type of gastric ulcer causes it to inherit 363698007 |finding site| = 78653002 |gastric mucosa| so it will now be included in the query for disease in the gastric mucosa.

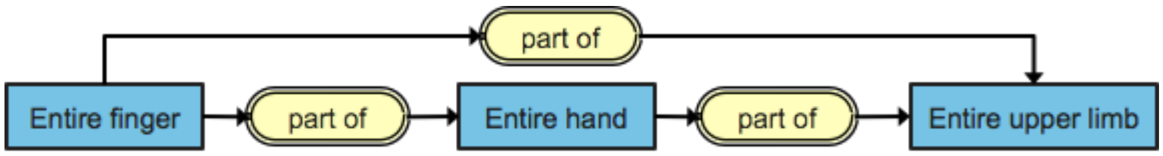
2.3.3 Additional Logic Features

Several description logic features, which would improve the completeness and precision of classification, cannot be represented in individual concept definitions.

Important Note
The examples on this page are illustrative only. Practical requirements and benefits of representing property transitivity and chains have been established. However, the concept model rules for applying these have not yet been finalized. Therefore, one of the two examples below refers to two concepts that do not currently exist in SNOMED CT and in the other example, it is likely that transitivity will apply to a specific subtype of the general |part of| attribute shown in the example.

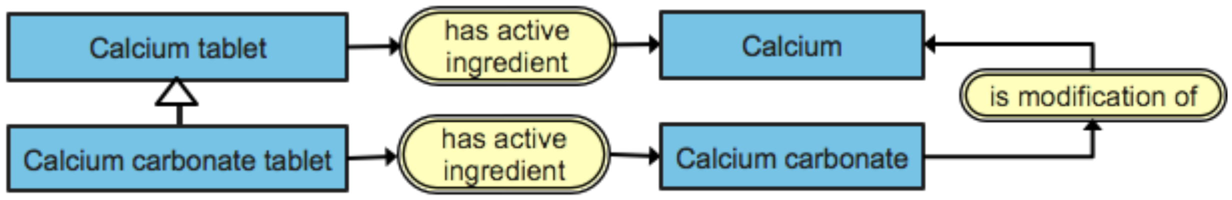
Property Characteristics

If particular characteristics of the attributes used to define concepts are identified, this can enhance classification. For example, knowing that an attribute like 123005000 |Part of| is transitive would allow the classifier to make infer that an entire finger is part of the entire upper limb.



Property Chains

Property chains are in some ways similar to transitivity but involve more than one attribute. For example, 127489000 |Has active ingredient| could be declared to chain with 738774007 |Is modification of|. In that case, suppose the definition of 387307005 |Calcium carbonate| includes 738774007 |Is modification of| of 5540006 |Calcium|, a DL classifier can determine that a tablet that 127489000 |Has active ingredient| = 387307005 |Calcium carbonate| by definition has a modification of 5540006 |Calcium| as its active ingredient. Thus in the example shown the classifier can determine that |calcium carbonate tablet| is a subtype of |calcium tablet|.



2.3.4 Representing Concept Definitions

This section summarizes the ways in which SNOMED CT concept definitions are represented. It outlines the rationale for changes to the way in which the stated view of concept definitions is represented. The changes are occurring during a transitional period that started in July 2018 and is due to be completed in 2019.

2.3.4.1 Concept Definitions Represented as Relationships

Between the first release of SNOMED CT in 2002 and 2018 both stated and inferred views of concept definitions were distributed as defining relationships in the stated relationship file and the relationship file.

Glossary Definition
A defining relationship is a relationship to a target concept that is always necessarily true for any instance of the source concept.

Notes

- All *defining relationships* represent necessary conditions. However, some necessary conditions that can be represented by OWL Axioms cannot be represented by *relationships*.

Example

- The *defining relationships* of the concept 53442002 |gastrectomy| include
  - 260686004 |method| = 129304002 |excision - action| and
  - 405813007 |procedure site - Direct| = 69695003 |stomach structure|.

As illustrated in figure 2.3.4.1-0, each defining relationship is represented by a row in the relationship file. The concept being defined is referenced by the sourceId, the concept that represents the type of relationship (attribute) is referenced by the typeId and the destinationId refers to the concept that represents the value of that attribute.

The relationship file also has a relationshipGroup which allows two or more defining relationships to be grouped together.

The definitionStatusId of the source concept, indicates whether the combination of defining relationships provide provides sufficient definition of that concept.

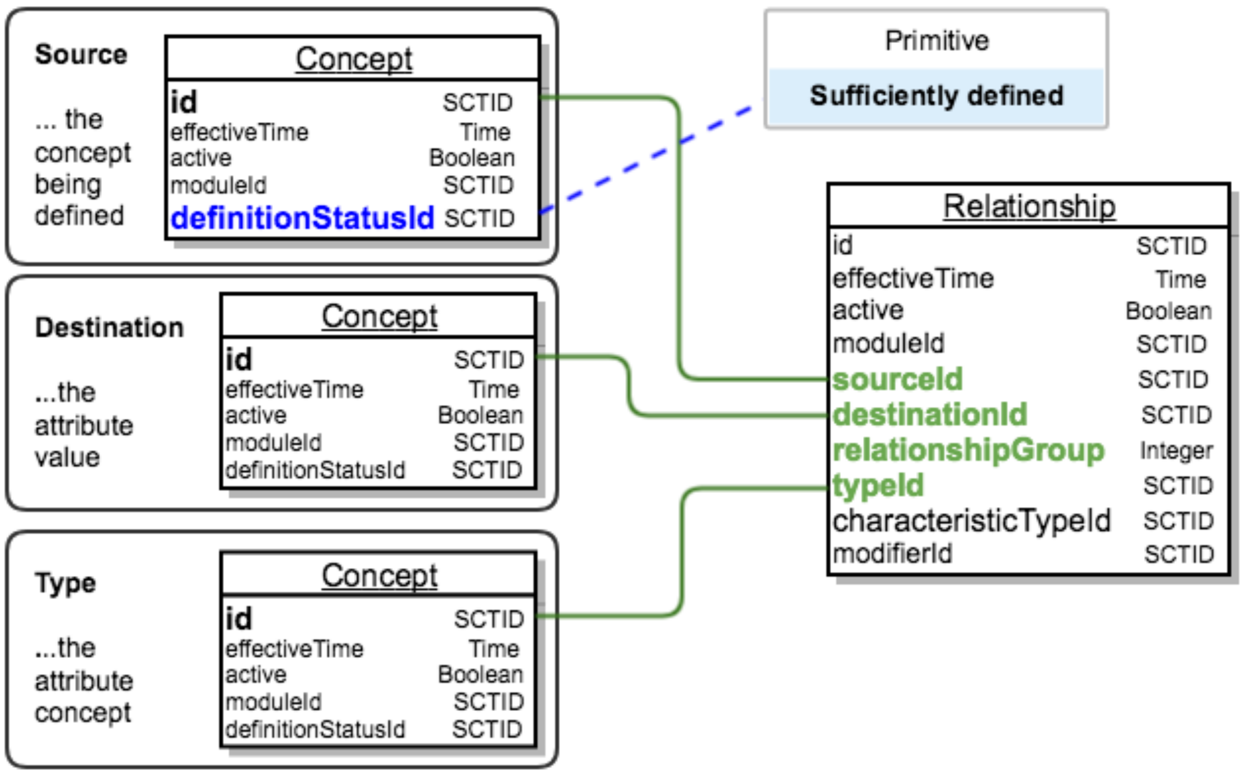


Figure 2.3.4.1-1: Diagrammatic representation of use of relationships to represent a concept definition

Table 2.3.4.1-1, shows the three rows in the relationship file that represent the definition of 53442002 |Excision of stomach structure|. As this is considered to be a sufficient definition of |Excision of stomach structure| the definitionStatusId of this concept is set to the value 900000000000073002 |defined|.

Table 2.3.4.1-1: Example of stated view of |gastrectomy| represented by stated relationships

	sourceId	destinationId	relationship Group	typeId	characteristicTypesId
	53442002 |Excision of stomach structure|	116680003 |Is a|	0	71388002 |Procedure|	900000000000010007 |Stated relationship|
	53442002 |Excision of stomach structure|	260686004 |Method|	1	129304002 |Excision - action |	900000000000010007 |Stated relationship|
	53442002 |Excision of stomach structure|	405813007 |Procedure site - Direct|	1	69695003 |Stomach structure|	900000000000010007 |Stated relationship|

Limitations of Relationships for Representing Concept Definitions

Section 2.3.2 Necessary Conditions and Sufficient Definitions, illustrated the following three points, which are not supported by the current use of relationships to represent concept definitions:

1. A concept may have more than one sufficient definition.

  - Use of relationships only supports representation of a single sufficient definition for each concept. If a concept is marked as sufficiently defined, all it relationships are considered to be part of its sufficie nt definition.
2. A concept may have a sufficient definition that includes some assertions that are not necessary conditions
  - Relationships are all assumed to be necessarily true.
3. Some necessary conditions may not be part of a sufficient definition.
  - Including these additional necessary conditions may cause some valid subtypes concepts (or expressions) to be omitted from the results of classification.

Section 2.3.3 Additional Logic Features, identifies other useful features that are supported by description logic tools but cannot be represented using only SNOMED CT.

Benefits of Relationships for Representing Concept Definitions

Relationships can be distributed in an easy to understand relational file structure. The relationship file has been an established part of the standard set of SNOMED CT release files since the first release in 2002, with a revision in 2011-2012 to use RF2 to enhance versioning capabilities. Relationships can be retrieved, displayed and processed using widely understood techniques such as SQL making it easy to join the relationships to the concepts to which they relate.

Future Use of Relationships for Representing Concept Definitions

Stated Relationships to be Deprecated

The stated view of concept definitions needs to be enhanced to allow more flexible and expressive use of description logic. The structure of the relationship file is not suitable for this and a decision has been made to adopt the OWL Functional Syntax so that new DL features can be added over time. As a result, at the end of the current transition period (during 2019), update, the stated relationship file with be deprecated.

Information about the new representation for the stated view is included in section 2.3.4.2 Concept Definitions Represented in OWL.

Impact Assessment
This change only impacts people who use the current stated view. Proper use of the stated view requires access to and use of a description logic classifier. Most DL classifiers require data to be provided in a OWL format, so these users typically transform from the stated relationship file to OWL prior to use. The new SNOMED CT OWL Toolkit makes it easy to prepare a full OWL file for classification from current and new distribution formats.
Overall impact is expected to be low with significant benefits.

Relationships Used for Inferred View Only

The current relationship file will continue to be released containing the inferred view. Due to limitations of the relationship file format, the inferred definitions will not contain the more sophisticated DL features.
The relationship file:

- will only contain necessary conditions
- it will not distinguish between multiple sufficient definitions
- it will whether each necessary condition is part of any of the sufficient definitions.

Nevertheless, the end result will still be a more complete and precise than the current content of this file. The reason for this is that the inferred relationships in the file will be be generated by processing the enhanced stated view. Details of the way the inferred relationship are generated from the stated view are document in 2.5. Generating Necessary Normal Form Relationships from the OWL Refsets.

Impact Assessment
The limitation of this format should not impact the vast majority of users of this file. The inferred relationship file will continue to support subsumption testing of precoordinated concepts. The inferred relationship file, however, will no longer support the testing of subsumption of postcoordinated expressions. Accurate tests for subsumption of postcoordinated expressions will be possible using a DL classifier with the stated OWL axioms. Optimizations such as the use of preclassified expression repositories can still be used to assist run time subsumption testing.
Overall impact is expected to be low with significant benefits.

1 Some columns omitted: id, effectiveTime, active, moduleId and modifierId.
2 Id columns are shown with the term expanded for clarity.

2.3.4.2 Concept Definitions Represented in OWL

This section outlines the rationale for distributing an OWL representation of the stated view of concept definitions and provides an overview of the way OWL axioms are represented in SNOMED CT release files. More detailed information is published separately in the SNOMED CT OWL Guide and the SNOMED CT Logic Profile Specification.

Rationale for Using OWL

The mismatch between the requirements for representing enhanced concept definitions and the capabilities of the current stated relationship file might in theory be addressed by addition of columns to the file or adding additional information in reference sets. However, in practice this would create a more complex solution able to support a specific set of enhanced features.

Adopting the well-established OWL standards formats offers a more flexible solution that can represent the full range of description logic features. This approach enables SNOMED Internation to specify a particular logic profile to be applied to current releases of SNOMED CT, with the option to extend that profile in the future. Future revisions of the logic profile would not require a change in the distribution file structure provided these were supported by an OWL syntax.

OWL Axioms

OWL axioms can be represented using several different syntaxes. SNOMED International has chose the OWL Functional Syntax as its standard representation.

The OWL Functional Syntax is a formal representation of the web ontology language (*OWL*) as a simple text base syntax that is used as a bridge between the structural specification and various concrete syntaxes.

Related Links

- OWL Functional-Style Syntax Specification

OWL Expression Axiom Set

OWL axioms are distributed in a reference set that follows the OWL Expression Reference Set specification. The axiom itself is contained in a string field and the concept whose definition it contributes to is referenced by the the referencedComponentId. Although a single row in the reference set can provide a sufficient definition, the definition of a single concept can also include several axioms each represented by a row in the reference set.

Comparing Stated Relationships and OWL Axioms

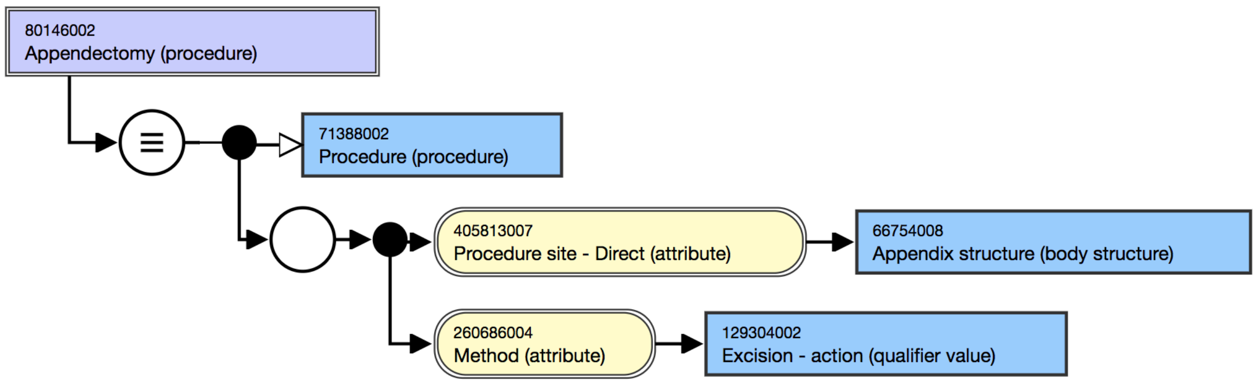


Figure 2.3.4.2-1: Diagrammatic representation of the definition of appendectomy

figure 2.3.4.2-0 shows the diagrammatic representation of the stated view of the definition of 80146002 | Appendectomy|. Table 2.3.4.2-1 shows the same definition as represented by three rows in the stated relationship file together with the definitionStatusId in the concept file.

Table 2.3.4.2-1: Stated relationships and definition status for the concept appendectomy

id	effectiveTime	active	..	definitionStatusId
80146002 |Appendectomy|	20020131	1	..	900000000000073002 |Sufficiently defined concept|

id	effective Time	active	..	sourceId	destinationId	relationship Group	typeId	..	..
..	20180731	1	..	80146002 |Appendectomy|	71388002 |Procedure|	0	116680003 |Is a|	..	..
..	20080731	1	..	80146002 |Appendectomy|	129304002 |Excision - action|	1	260686004 |Method|	..	..
..	20080731	1	..	80146002 |Appendectomy|	66754008 |Appendix structure|	1	405813007 |Procedure site - Direct|	..	..

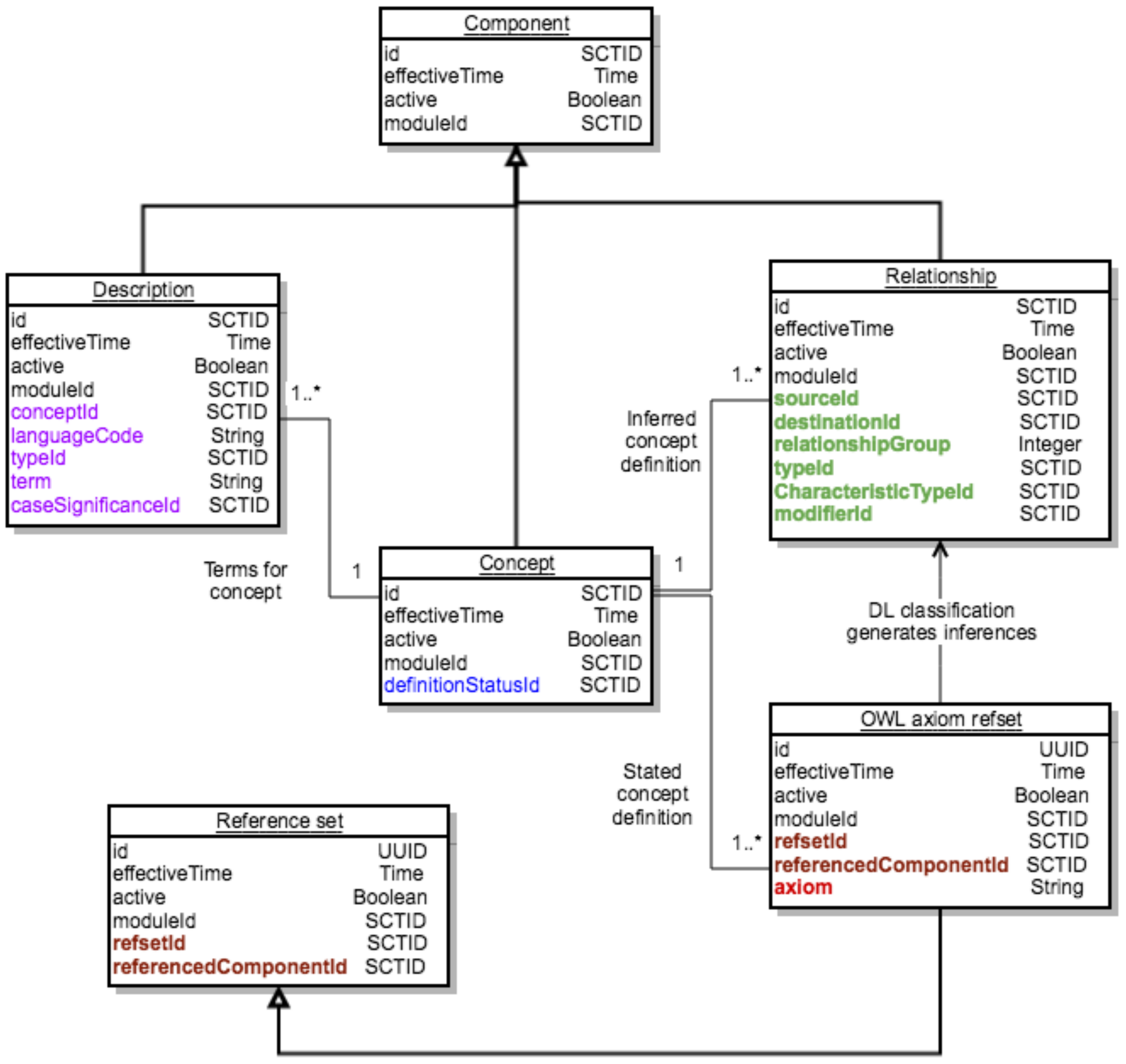
Table 2.3.4.2-2 shows a row in the OWL axiom reference set file representing the same definition. As shown by this example, a single sufficient definition is represented by a single row in the reference set. However, some concept definitions may require multiple rows in the reference set. Situations in which multiple row are required include:

- Concepts with multiple sufficient definitions, each of which requires a separate row in the reference set.
- Concepts with additional necessary conditions that are not part of a sufficient definition, each of which requires a separate row in the reference set.

Table 2.3.4.2-2: Example of OWL axiom refset representation of the definition of appendectomy

id	effective Time	active	moduleId	refsetId	referenceComponentId	owlExpression

..	🗎	1	..	733073007 |OWL axiom reference set|	80146002 |Appendectomy|	EquivalentClasses(:80146002 ObjectIntersectionOf(:71388002 ObjectSomeValuesFrom(:609096000 ObjectIntersectionOf(ObjectSomeValuesFrom(:260686004 :129304002) ObjectSomeValuesFrom(:405813007 :66754008)))))



3 Release Types, Packages and Files

This section covers several general topics related to SNOMED CT release packages and release files including:

The following two sections provide detailed specifications of the release files.

- Section 4 Component Release Files Specification contains details of release files that represent the main SNOMED CT components:
  - Concepts
  - Descriptions
  - Relationships
- Section 5 Reference Set Release Files Specification contains details of release files that customize and enhance SNOMED CT by representing:
  - Subsets of concepts and descriptions
  - Maps to other code systems
  - Language and dialect preferences for different terms
  - Annotation of components
  - Associations between components
  - Other forms of configuration and extensibility

Historical Note on Current and Previous Release File Formats
The standard format in which SNOMED CT has been distributed since 2012 is known as Release Format 2(RF2). It was developed in response to extensive feedback on the original release file format, now known as Release Format 1(RF1), in which SNOMED CT was distributed between its first release in 2002 and 2012. The RF1 format is now deprecated and no longer supported.

3.1 Common Features of All Release Files

This subsection explains features that apply to all SNOMED CT release files.

3.1.1 General Structure of Release Files

The following rules apply to all SNOMED CT Release Files.

- SNOMED CT Release Files are UTF-8 encoded, tab delimited text files.
- Each line, including the final line, ends with a carriage return character (hex 0D) followed by a line feed character (hex 0A).
- The first line of each file, contains the names of each column (also know more generally as a field).
- Field names are represented using lower-camel-case
  - First letter of name is lower case
  - First letter of each words apart from the first word in the name is upper case
  - All other letters are lower case.
  - For example:
    - id
    - term
    - typeId
    - relationshipGroup
    - definitionStatusId
- The name, datatype and usage of the fields in each file are specified in the following sections of this guide
  - 4.2 File Format Specifications
  - 5.2 Reference Set Types.

3.1.2 Release File Data Types

The following data types are used in the release files:

Table 3.1.2-2: Data Types Used in Release Files

Data Type	Description
SCTID	A SNOMED CT identifier, between 6 and 18 digits long, as described in 6.2 SCTID Representation. • This data type is used to identify SNOMED components, to refer to a component from another component or from a reference set, and also to represent the values for concept enumerations (see Concept Enumerations.).
UUID	A Universally Unique Identifier is a 128-bit unsigned generated using a standard algorithm. • UUIDs are represented as strings of hexadecimal characters split by - characters as points specified by the UUID standard.
Integer	A 32-bit signed integer.
String	UTF-8 text of a specified length.
Boolean	A Boolean value, represented as one of two possible integer values (1 = true, 0 = false).

Time	A date and time format expressed as a text string in line the basic representation specified in the ISO 8601 standard. • Where only date is required the format is YYYYMMDD (e.g. 20180125 refers to 25th January 2018) • Where a time is also required the *YYYYMMDDThhmmss* Z (e.g. 20180125T123000Z refers to 12:30 UTC on 25th January 2018) • The time should be expressed as UTC, as indicated by the trailing "Z".

Concept Enumerations

Concept enumeration is the a set of SNOMED CT concept identifiers used to represent values for a property of a SNOMED CT component or reference set member.

Notes

- *Concept enumeration* serves the same purpose as more general approaches to providing enumerated lists of values (i.e. assigning a number to a value). However, the use of SNOMED CT concept identifier allows access to the human readable meaning of each enumeration using descriptions in the same way for other concepts.
- The SNOMED CT concepts used to represent *concept enumerations* are usually subtype children (or descendants) of concepts in the SNOMED CT metadata hierarchy. Each possible value is represented by a single child concept. This allows updates to the permitted values to be tracked using the component history mechanism.

Example

- Concept enumerations for description.typeId:

```
900000000000446008 |Description type (core metadata concept)|
   900000000000003001 |Fully specified name (core metadata concept)|
   900000000000013009 |Synonym (core metadata concept)|
   900000000000550004 |Definition (core metadata concept)|
```

Table 3.1.2-2: Concept enumeration values (subtypes of 900000000000442005 | Core metadata concept |)

Concept	Comment
900000000000443000 |Module (core metadata concept)|	Each subtype of this concept represents a development module. These concepts provide values to the moduleId field that is present in all SNOMED CT component file. The value indicates the module within which a component was created and is being maintained.
900000000000444006 |Definition status (core metadata concept)|	Each subtype of this concept represents a value that can be applied to the concept. definitionStatusId field. This is used to indicate whether the current set of defining Relationships applied to a concept are sufficient to fully-define it relative to its supertypes.
900000000000446008 |Description type (core metadata concept)|	Each subtype of this concept represents a value that can be applied to the Description. typeId field. This is used to indicate whether the Description represents a Fully Specified Name, a synonymous term, a definition or some other symbolic or textual representation of the associated concept .
900000000000447004 |Case significance (core metadata concept)|	Each subtype of this concept represents a value that can be applied to the Description. caseSignificanceId field. This is used to indicate whether the text of the term can be modified to by switching characters from upper to lower case (or vice-versa).

Concept	Comment
900000000000449001 |Characteristic type (core metadata concept)|	Each subtype of this concept represents a value that can be applied to the Relationship. characteristicTypeId field. This is used to indicate whether a Relationship forms part of the definition of the source concept .
900000000000450001 |Modifier (core metadata concept)|	Each subtype of this concept represents a value that can be applied to the Relationship. modifierId field. This is used to indicate the type of Description Logic (DL) restriction (some, all, etc.) that applies to the Relationship .
900000000000453004 |Identifier scheme (core metadata concept)|	Each subtype of this concept represents a value that can be applied to the Identifier. identifierSchemeId field. This is used to indicate the scheme to which the Identifier value belongs.

3.1.3 Fields Present in All Release Files

The first four columns in all release files are shown in Table 3.1.3-1. The next three sections of the specification explain the ways in which these fields are used to support identification, versioning and modularization.

Table 3.1.3-1: Fields present in all release files

Field	Data type	Purpose	Mutable	Part of Primary Key
id	SCTID	Uniquely identifies a component.	NO	YES (Full/ Snapshot)
	UUID	Uniquely identifies a reference set member.		
effectiveTime	Time	Specifies the inclusive date at which this rows state became the then current valid state of the identified component or reference set member	YES	YES (Full) Optional (Snapshot)
active	Boolean	Specifies whether the state of the component or reference set member was active or inactive from the nominal release date specified by the effectiveTime .	YES	NO
moduleId	SCTID	Identifies the module this component or reference set member is currently maintained in. Set to a child of 900000000000443000 | Module | within the metadata hierarchy.	YES	NO

3.1.4 Meaning of the Active Field

Each component has an associated active field, which can take values of true (1) or false (0). The meaning of this flag is described by component type in the following table:

Table 3.1.4-1: Behavior of Active and Inactive Components

Component Type	Active **value**	Component **behavior when the most recent row representing a component has the specified active value**
Concept	True	• The concept is intended for active use. • All active descriptions for which the conceptId refers to this concept are valid. Visibility of these active descriptions depends on information contained in applicable refset members (for example, whether the description is in a language dialect reference set that is currently enabled in the vendor's system). • All active relationships of which it is the sourceId or destinationId are applicable.
Concept	False	• The concept is not intended for active use. However, it remains a valid concept for historical purposes as part of the SNOMED CT commitment to the principle of ' concept permanence'. • Valid descriptions of the concept remain active allowing it to be appropriately viewed in human-readable form. • An inactive concept cannot be the sourceId , destinationId or typeId of an active relationship .
Description	True	• The description contains a term that is a valid description of the concept referred to by the conceptId. • An active description may refer to an inactive concept , in which case the term provides a valid description of that inactive concept . Text based searches should (by default) include only active descriptions that refer to active concepts .
Description	False	• The description is not a valid and the associated term should no longer be regarded as being associated with the concept referred to by conceptId.
Relationship	True	• The relationship represents a valid association of the type specified by the typeId , between two concepts referred to by the sourceId and destinationId ; • An inactive concept cannot be the sourceId , destinationId or typeId of an active relationship .
Relationship	False	• The relationship is not valid. An inactive relationship should be ignored as it does not apply. • This does not necessarily mean that the association indicated by the relationship does not apply. The relationship may be inactive because it is redundant and inferable based on other active relationships . • An inactive relationship may refer to either active or inactive components .
Refset member	True	• The refset member contains valid information applicable to the component referred to by the referencedComponentId . • The component referred to by the referencedComponentId may be active or inactive . An active refset member cannot make an inactive component active but may provide related information that continues to be relevant (e.g. the reason for inactivation).
Refset member	False	• The refset member is not valid. An inactive refset member should be ignored. The information it contains is not applicable to the component referred to by referencedComponentId .

3.1.5 History Mechanism

The effectiveTime and active fields in the release file enable the use of a "log style" append-only data model to track all changes to each component, providing full traceability. Once released, a row in any of these files will always remain unchanged. Historic data is supplied in the RF2 release files, dating back to the first release in RF1 format in 2002.

In order to change the properties of a current component a new version of that component is created with the same identifier. This done by adding a new row to the relevant release file, with the column values updated to represent the changes. The active field must be set to true and the timestamp in the effectiveTime field indicating the nominal date on which the new version was released. Note that the existing row is not changed in any way.

To inactivate a component, a new row is added, containing the same data as the final valid version of the component, but with the active field set to false and the timestamp in the effectiveTime field indicating the nominal date of the release in which the final version ceased being valid. Note again that the existing row is not changed in any way.

Where editorial policy does not allow a particular property of a component to be changed whilst keeping the same Identifier, the component as a whole is inactivated (as described above), and a new row added with a new id, the effectiveTime set to the nominal date of the release in which this version of the component became valid, and the active field set to true.

It is thus possible to see both the current values and any historical values of a component at any point in time.

New content, changes and inactivations must have the effectiveTime for the release that it appears in. Pre-releases for testing may set the effectiveTime as the date of the future scheduled release but in general the effectiveTime must not be later that the scheduled release data, Where there is a business requirement for specifying a future activation date for some components, this may be represented using reference sets.

The following example demonstrates how the *history mechanism* works on the Concept, but the same rules apply equally well to the Description, Relationship and Reference set member files. In this example, the descriptions associated with the moduleId and definitionStatusId have been shown in place of their SCTID values.

A new concept (101291009) is added on the 1st July 2007:

Table 3.1.5-4: History Example - Concept Added

Id	effectiveTime	active	moduleId	definitionStatusId
101291009	20070701	1	|Module 1|	900000000000074008 |Primitive|

In the following release (on 1st January 2008), the concept is moved from |Module 1| to |Module 2|. Because the moduleId field is not immutable, the concept may be updated simply by adding a new record with the same Id.

Table 3.1.5-4: History Example - Module Change

Id	effectiveTime	active	moduleId	definitionStatusId
101291009	20070701	1	|Module 1|	900000000000074008 |Primitive|
101291009	20080101	1	|Module 2|	900000000000074008 |Primitive|

In the following release (on 1st July 2008), the concept is changed from being Primitive to being Fully defined.

Table 3.1.5-4: History Example - Definition Status Changed

Id	effectiveTime	active	moduleId	definitionStatusId
101291009	20070701	1	|Module 1|	900000000000074008 |Primitive|
101291009	20080101	1	|Module 2|	900000000000074008 |Primitive|
101291009	20080701	1	|Module 2|	900000000000073002 |Defined|

In the following release (on 1st January 2009), the concept is deactivated:

Table 3.1.5-4: History Example - Concept Made Inactive

Id	effectiveTime	active	moduleId	definitionStatusId
101291009	20070701	1	|Module 1|	900000000000074008 |Primitive|
101291009	20080101	1	|Module 2|	900000000000074008 |Primitive|
101291009	20080701	1	|Module 2|	900000000000073002 |Defined|
101291009	20090101	0	|Module 2|	900000000000074008 |Primitive|

Notes

1. At no stage in this process are previously written records ever amended. Once a record has been released in a release file, it will continue to be released in exactly the same form in future release files.
2. Changes are only recorded at the point of release in the RF2 release files. If a component record is changed a number of times between releases (during an edit and review process), only the most recently amended record will be appended to the release file, not individual records showing each separate edit to the released component.
3. In the last example, as well as inactivating the concept (active=0), the definitionStatusId is changed from 900000000000073002 |Defined| to 900000000000074008 |Primitive|. In practice this change is not essential since the value of data columns is ignored when a component is inactive. Although the change is unnecessary and insignificant, it typically occurs since all the relationships of an inactive concept must also be inactive, and as a result, from the perspective of the authoring environment the concept cannot be regarded as 900000000000073002 |Defined|.

Related Links

- 3.1.4.1. Component features - History

3.1.6 Module Identification

Each SNOMED CT component is managed and maintained in a module identified by its moduleId field.

A SNOMED CT module is defined as a group of SNOMED CT components and/or reference set members managed, maintained, and distributed as a unit.

Notes

- Components and reference set members that are part of the same *module* share the same moduleId value.
- All *modules*, except the 900000000000012004 |SNOMED CT model component module|, have dependencies on other *modules* specified by the Module Dependency Reference Set.
    - components and reference set members, that are part of the same *module,* share the same moduleId value.
    - components and reference set members are part of only one *module*, at any given time.
- The organization responsible for maintaining an extension must:
    - create and maintain at least one *module* identified by a *moduleId* that it has created;
    - apply a *moduleId* that it has created to all components and reference set members in its *extension*;
    - manage and distribute information about the dependencies of its *modules* in accordance with SNOMED CT specifications.
- The organization responsible for maintaining an extension may:
    - create and maintain multiple *modules*;
    - organize its components and reference set members within the modules it manages in a way that best meets its business needs;

    - move a component or reference set member between its *modules* by creating a revised version of that component or reference set member with a different *moduleId* (It is then part of the new *module* from the *effectiveTime* of the revised version).
- Components and reference set members may be moved between *modules* maintained by different organizations. However, such moves must only be made:
    - with the consent of the organizations responsible for both the source and target modules; and
    - in accordance with rules specified by SNOMED International.

3.2 Release Types

A SNOMED CT International Release includes three distinct release types. Table 3.2-1 describes the release types and the differences between them.

Table 3.2-1: SNOMED CT Release Types

Release type	Description
Full	A full release is a release type in which the release files contain every version of every component and reference set member ever released.
Snapshot	A snapshot release is a release type in which the release files contain only the most recent version of every component and reference set member released, as at the release date.
Delta	A delta release is a release type in which the release files contain only rows that represent component versions and reference set member versions created since the previous release date. Notes • Each *row* in a *delta release* file represents either a new component or reference set member, or a change to an existing component or reference set member since the previous release date. • A *delta release* identifies differences between two versions of the same release package. • A *delta release* added to the previous full release is identical to the full release of the new version. • The previous *release date*, on which a *delta release* is based, is usually the date of the most recent previous release. However, that may not always be the case. For example, where interim releases are made between two major releases there may be a combined *delta release* covering a period since a previous major release.

There are practical use cases for each Release Type.

- The full release allows access to all versions of the release. This is valuable for reviewing data entered using earlier versions and more generally supporting change management.
- The snapshot release only includes the latest version of each component. This can be useful to optimize access to the current version but does not provide access to earlier versions.
- The delta release only includes changes made between one version and the next. This provides a simple way to identify new and changed components to support change management and can also be used to update the previous version of the full release to the new version of the full release. However, the delta release cannot be used as a stand alone resource.

When considering which release type to use, it is worth noting that delta and snapshot views can be readily generated from the full release type. For this reason organizations that maintain SNOMED CT extensions are required to provide the full release type, while distribution of the other release types are optional.

3.3 Naming Conventions for Release Packages and Files

The release file naming convention specified in this section applies to all SNOMED International release files starting with the January 2010 International release. The release package naming convention specified was added subsequently to provide additional clarity through a structured name applied to the folder (and zip archive name) containing a set of release files. This package naming convention first applied to releases by SNOMED International during 2017.

These naming specifications provides the following benefits:

- A consistent naming convention across the International edition and each National edition.
- Predictable file naming, providing a stable structure for naming over time between releases.
- A standard way to identify the source country and namespace of the organization responsible for a release file.
- A consistent mechanism for representing version of release files and packages of release files.
- An human readable way to identify the content of a file, at a summary level.
- A mechanism for identifying the type of information stored in a release file(e.g. documentation, tooling, etc.).
- Guidance on file naming for release files in non-English extensions.
- Assurance that file names will be unique across the International release and releases from individual National release centers and across separate releases from each center over time.

Quality Assurance checks, performed during the International release process, ensure that this naming convention is enforced. Equivalent checks should also be performed as part of each National Release Center's release process.

Note
Prior to January 2010 other naming conventions were used. Implementers who need to review earlier releases should consult the documentation that accompanied the release that they need to review

3.3.1 Release Package Naming Conventions

Overall Package Naming Pattern

```
SnomedCT_[Product][Scope(optional)][Format(optional)]_[ReleaseStatus]_[ReleaseDate]T[R
eleaseTime][TimeZone]
```

Package Name Elements

Element	Values	Description
Product	<any>	Camel case short title sufficient to identify the product.
Scope (optional)	Edition	The release files included in the package fully resolve all dependencies of all modules included in the package.
	Extension	The release files included in the package needs to be combined with the International Edition release package and any other packages required to resolve the dependencies declared by the Module Dependency Reference Set .
Format (optional)	RF1	Required for any release packages containing Release Format 1 files.
	RF2	Current value for all release packages.
	<other>	Other values may be specified in future.
ReleaseStatus	ALPHA	The package is an alpha release package , which is defined as a SNOMED CT release package made available only for initial review and testing by implementers and other stakeholders. Notes • An *alpha release package* must not be used in production clinical systems or in clinical settings. This includes Affiliate Licensees or any third parties, except those who have formally committed to test it. • An *alpha release* is used to test the format and content of the SNOMED CT release . Feedback is elicited and changes are made prior to publication of the beta release . • *Alpha releases* were formerly known as a technology preview releases.
	BETA	The package is a beta release package , which is defined as a SNOMED CT release package made available for review and testing only. Notes • Implementers and other stakeholders review and test the *beta release* . • The *beta release package* is made available prior to the production release . It must not be used in production clinical systems or in clinical settings. This includes Affiliate Licensees or any third parties, except those who have formally committed to test it. • The *beta release* status indicates it is expected to subsequently be confirmed as a production release . If there is significant issue in format or content, it may be withdrawn, or replaced with an updated *beta release package* . Whether or not it becomes a production release is decided shortly before the due date for the next release. If a *beta release* is subsequently confirmed as a production release , all updates are fully version-tracked from the date of the *beta release* . • Beta releases were formerly known as candidate baseline releases.
	PRODUCTION	The package is a production release package , which is defined as a final, formally endorsed SNOMED CT release package intended for live use in appropriately licensed operational systems. Notes • A *production release package* represents the authoritative release of the product. Implementers can use it in operational clinical systems. • The *production release* status indicates that the releasing party (SNOMED International or the owner of the extension) commits to maintain the release history. Thus the historical audit trail is maintained through the product's lifetime.

ReleaseDate	YYYYMMDD	The package release date, time and timezone formatted in accordance with ISO-8601 .
ReleaseTime	HHMMDD	
TimeZone	Z	

3.3.2 Release File Naming Convention

Overall Naming Pattern

The basic pattern for SNOMED CT release file names consists of five elements, each separated by an underscore (" _ ") and followed by a full stop (" . ") and a file extension:

```
[FileType]_[ContentType]_[ContentSubType]_[CountryNamespace]_[VersionDate].[FileExten
sion]
```

Each element in the above structure is described in more detail by table in the following section.

FileType Element

The FileType element of the filename designates the type and intended use of the release file . It consists of a 3 to 5 alphanumeric code with letters in lowercase.

The code comprises the following three sub-elements. The Type sub-element is required in all cases, other elements are required where relevant and otherwise omitted.

Table 3.3.2-9: FileType Element - Sub-elements and Permitted Values

Sub-element	Values	Description
Status	<blank>	General release file
	x	Provisional release file (e.g. part of an alpha or beta release package).
	z	Archival or unsupported file
Type	sct	Terminology Data File
	der	Derivative Work Data File (e.g. Reference set release file)
	doc	Documentation
	res	Implementation Resource Data File (e.g. a data file not following a SNOMED CT standard release file format)
	tls	Implementation Resource Tool (e.g. scripts or other software made available to process a release file)
Format	1	Release Format 1
	2	Release Format 2
	<blank>	Not specific to a release version

ContentType Element

The ContentSubType element is mandatory for all FileTypes. It describes the content and purpose of the file. It consists of 2-48 alphanumeric characters in camel case.

The content of this element depends on the first element (FileType) of the filename, as described below:

Table 3.3.2-9: ContentType Element - Permitted Values for FileType "sct"

Value	Usage
Concept	The file conforms to the 4.2.1 Concept File Specification and contains data related to a set of concepts.
Relationship	The file conforms to the 4.2.3 Relationship File Specification and contains relationships that represent the distribution normal form inferred view of a set of concept definitions.

sRefset	The file conform to the single string reference set format. This only applies to the OWL Expression Reference Set and followed by the content sub-element _OWLExpression which contains stated concept definitions represented as OWL axioms and additional OWL ontology information.
Description	The file conforms to the 4.2.2 Description File Specification and contains at set of descriptions with description types | Synonym | and | Fully specified name | . Note that both these description types have a maximum term length of 255 characters.
TextDefinition	The file conforms to the 4.2.2 Description File Specification and contains at set of descriptions with description type . Note: This description type has a maximum term length of 4096 characters.
StatedRelationship	The file conforms to the 4.2.3 Relationship File Specification and contains relationships that represent the stated view of a set of concept definitions. Note: It is likely this file will be phased out and replaced with a reference set containing a richer OWL representation of stated concept definition.
Identifier	The file conforms to the 4.2.4 Identifier File Specification . Note: This file does not contain any data rows in the International Edition.

Table 3.3.2-9: ContentType Element - Permitted Values for FileType "der"

Value	**Description**
Refset	The file conforms to the 5.2.1 Simple Reference Set specification and contains the members of one or more simple reference sets.
<pattern> Refset	The file conforms to the 5.1.1 Basic Reference Set Member File Format and include one or more additional columns, The number and order of the columns and their basic data types are specified by the <pattern> which precedes Refset. The <pattern> consists of a sequence of lowercase letters each of which represent an additional column with a datatype specified by the letter as listed below
Pattern letter	
c	A SNOMED CT component identifier (SCTID) referring to a concept, description or relationship.
i	A signed integer.
s	A UTF-8 text string.
Examples	• **cRefset** : A refset with one additional column containing a component identifier. This pattern supports refset types including: 5.2.3 Attribute Value Reference Set , 5.2.4 Language Reference Set and 5.2.5 Association Reference Set). • **ciRefset** : A refset with two additional columns, one containing a component identifier and one containing an integer. This pattern supports refset types including: 5.2.6 Ordered Association Reference Set . • **sRefset** : A refset with one additional column containing a string. This pattern supports refset types including: 5.2.9 Simple Map Reference Set and 5.2.7 Annotation Reference Set .

Table 3.3.2-9: ContentType Element - Permitted Values for FileTypes "doc","res" and "tls"

FileType	**Value and Description**

doc	The title of the document in CamelCase, abbridges if necessary to fit within the length constraint. Note: Abbreviations should not be used unless they are essential to fit the title within the available length. **Examples of ContentType for Documents** • doc_ **SnomedDecisionSupport** _Current-en-US_INT_20170331.pdf (Title: Decision Support with SNOMED CT) • doc_ **SearchDataEntryGuide** _Current-en-US_INT_20171122 (Title: SNOMED CT Search and Data Entry Guide)
res tls	The value of the ContentType element may be determined on a case-by-case basis but, in conjunction with the ContentSubType element, should be adequate to identify the content and purpose of the file.

ContentSubType Element

The ContentSubType element is mandatory for all FileTypes. It provides additional information to describe the content and purpose of the file, including the language/ dialect, where appropriate. Its format is 2-48 alphanumeric characters in camel case (except for the capitalization rules specified below for languagecode). Hyphen (" **-** ") is a permitted character in conjunction with a language code, as described below.

Table 3.3.2-5: ContentSubType Element - Sub-elements and Permitted Values for FileTypes "sct" and "der"

Sub-elements	Values	Description
Summary		An optional short camel case summary of the usage of the file. The value of this sub-element may be determined on a case-by-case basis but, in conjunction with the ContentType element, should be adequate to identify the content and purpose of the file. Examples: • For references sets a brief indication about the type or purpose the reference set(s) in the file. Note: If there is a summary the ReleaseType or DocStatus follows this Summary sub-element immediately without a space or other separator.
ReleaseType	Full	The file contains the Full view of the components or refset members within its scope (i.e. every version ever released).
	Snapshot	The file contains the Snapshot view of the components or refset members within its scope (i.e. only the most recent version released).
	Delta	The file contains the Delta view of the components or refset members within its scope (i.e. only additions/ changes since previous release).
LanguageCode		Where it is necessary to specify the language or dialect used in a file, the appropriate language code must be included as the final sub-element of the ContentSubType. If a Summary or DocStatus sub-element is also included, the LanguageCode must be added after the last of those sub-elements and must be separated from it by a hyphen. **Representation of the LanguageCode** The language is specified with a 2 character ISO 639-1 language code (e.g. es = Spanish, fr = French, da = Danish). If necessary, a dialect code is added after the langauge code and seperated from it by a hyphen. Depending on the specificity required the dialect code comes from one of two sources: • If the dialect is general to an entire country, the two-letter ISO-3166 alpha-2 country code is used to specify the dialect (e.g. en-US = US English, en-GB British English) • If dialect is less common or not country specific, the IANA language subtag should be used. Note this code consists strings of lower case letres letters. IANA is the Internet Assigned Numbers Authority. This approach follows Internet conventions.

Table 3.3.2-9: ContentSubType Element - Sub-elements and Permitted Values for FileType "doc"

Sub-elements	Values	Description
Summary		An optional short camel case addition to the ContentType title. If there is a Summary the DocStatus follows this Summary sub-element immediately without a space or other separator.
DocStatus	Current	The document is up-to-date and complete for the current release of SNOMED CT, as indicated by the VersionDate element.
	Draft	The document is a draft version; it may be incomplete and has not been approved in a final version.
	Review	The document has been released for review and comments from SNOMED International Members, Affiliates and other stakeholders.
LanguageCode		Where it is necessary to specify the language or dialect used in a file, the appropriate language code must be included as the final sub-element of the ContentSubType. If a Summary or DocStatus sub-element is also included, the LanguageCode must be added after the last of those sub-elements and must be separated from it by a hyphen. Representation of the LangageCode is described in detail in the final row of Table 3.3.2-5 .

Table 3.3.2-9: ContentSubType Element - Sub-elements and Permitted Values for FileTypes "res" and "tls"

Sub-elements	Values and Description
Summary	The value of this sub-element may be determined on a case-by-case basis but, in conjunction with the ContentType element, should be adequate to identify the content and purpose of the file.
LanguageCode	If it is necessary to specify the language or dialect used in a resource data file or tool, the appropriate language code must be included as the final sub-element of the ContentSubType. If a Summary sub-element is also included, the LanguageCode must be added after the Summary sub-element and must be separated from it by a hyphen. Representation of the LangageCode is described in detail in the final row of Table 3.3.2-5 .

Examples of ContentSubType

- der2_cRefset_**AttributeValueSnapshot**_INT_20180131.txt
  - Summary=AttributeValue (type of refset),
  - Release type=Snapshot,
  - Language not stated
- sct2_Description_**Snapshot-en**_INT_20180131.txt
  - Release type=Snapshot,
  - Language=English
- der2_cRefset_**LanguageSnapshot-en**_INT_20180131.txt
  - Summary=Language (type of refset),
  - Release type=Snapshot,
  - Language=English
- doc_IhtsdoGlossary_**Current-en-US**_INT_20170817.pdf
  - DocStatus=Current,
  - Language=en-US.

CountryNamespace Element

The CountryNamespace element is mandatory for all FileTypes. It identifies the organization responsible for developing and maintaining the file. It is a string of 2 to 10 alphanumeric characters consisting of the two sub-elements described below. At least one of these two sub-elements must be present. SNOMED International or a

National Release Center (NRC) may optionally include both sub-elements where they consider this to be appropriate.

Table 3.3.2-9: CountryNamespace Element - Sub-elements and Permitted Values

Sub-element	Values	Description
CountryCode	INT	The file is maintained and distributed by SNOMED International.
	AA *to* ZZ	The file is maintained and distributed by the NRC for the country represented by this ISO-3166 alpha-2 country code . The code consists of exactly two uppercase characters from the latin alphabet.
	<blank>	The file is maintained and released by an SNOMED CT extension provider that is not an NRC.
NamespaceId	0000000 *to* 9999999	The file is maintained and released by an SNOMED CT extension provider that is not an NRC. In which case, this value is a 7 digit namespace identifier allocated to that organization by SNOMED International. The file is maintained and distributed by either SNOMED International or an NRC and the distributing organization has chosen to include the namespace identifier to indicate that this is part of a release restricted to content in a single namespace.
	<blank>	The file is maintained and distributed by either SNOMED International or an NRC and the distributing organization has not chosen to include the namespace identifier to indicate that this is part of a release restricted to content in a single namespace.

VersionDate Element

The VersionDate element is mandatory for all FileTypes. It identifies the SNOMED CT version with which the file is intended to be used. Its format is an 8-digit number in the pattern "YYYYMMDD", in compliance with the ISO-8601 standard.

- For Data Files(**sct**,**der** or **res**),and for Documentation (**doc**) with a *status* tag value of "**Current**", the value of this element should always be the same as the SNOMED CT version date with which the file is associated.
- For other file types, the VersionDate element will identify the (past) date of the SNOMED CT release for which the file was intended. A file distributed with a past version date has not been updated to reflect changes to SNOMED CT since that date, nor has it been validated as correct or appropriate for current use.

File Extension

The extension element of the filename identifies the file format (encoding convention) of the file, such as " **txt** ", " **pdf** " or " **zip** ". It has a format of 1-4 alphanumeric characters.

Table 3.3.2-9: File Extensions Applicable to Different FileTypes

FileType	Values	Description
sct or der	txt	All RF2 formatted release files are distributed as plain text UTF-8 files with the .txt suffix.
doc	pdf	Portable Document Format is the default format for documents distributed and made available for download in a format suitable for local viewing or printing.
	<other>	Other document formats including plain text (.txt) and HTML (.html) may be used where deemed appropriate. In all cases the file extension (suffix) used should be one of the widely recognized format. Unless there are exceptional requirements, the format should be accessible using freely available software.
res	txt	Most resources should be provided as plain text UTF-8 files with the .txt suffix.
	zip	Where appropriate a resource file, or a collection of such files, may be distributed as zip archive.
	<other>	Other data formats may be used where appropriate.
tls	<any>	No specific statements are made about the file extsions to be used for tooling files. However, in general such tools should be provided in a format that does not compromise system security. In most cases, tools should be provided through an interface such as GitHub and should not be included as part of general releases of the terminology.

3.4 Release Package Contents

This subsection provides illustrated notes on the contents of the SNOMED CT International release package used to distribute the SNOMED International Edition. Similar folder structures should be present in other release packages. However, the files included may be limited to those required to represent the components and/or reference set members in that particular release package.

figure 3.4-0 shows the overall structure of a release package. The top level folder is named according to the Release Package Naming Conventions. It contains one subfolder for each of the Release Types (Delta, Full and Snapshot). Each release type folder contains a Terminology folder and a Refset folder. The Refset folders contains separate folders for different groups of reference sets (Content, Language, Map and Metadata).

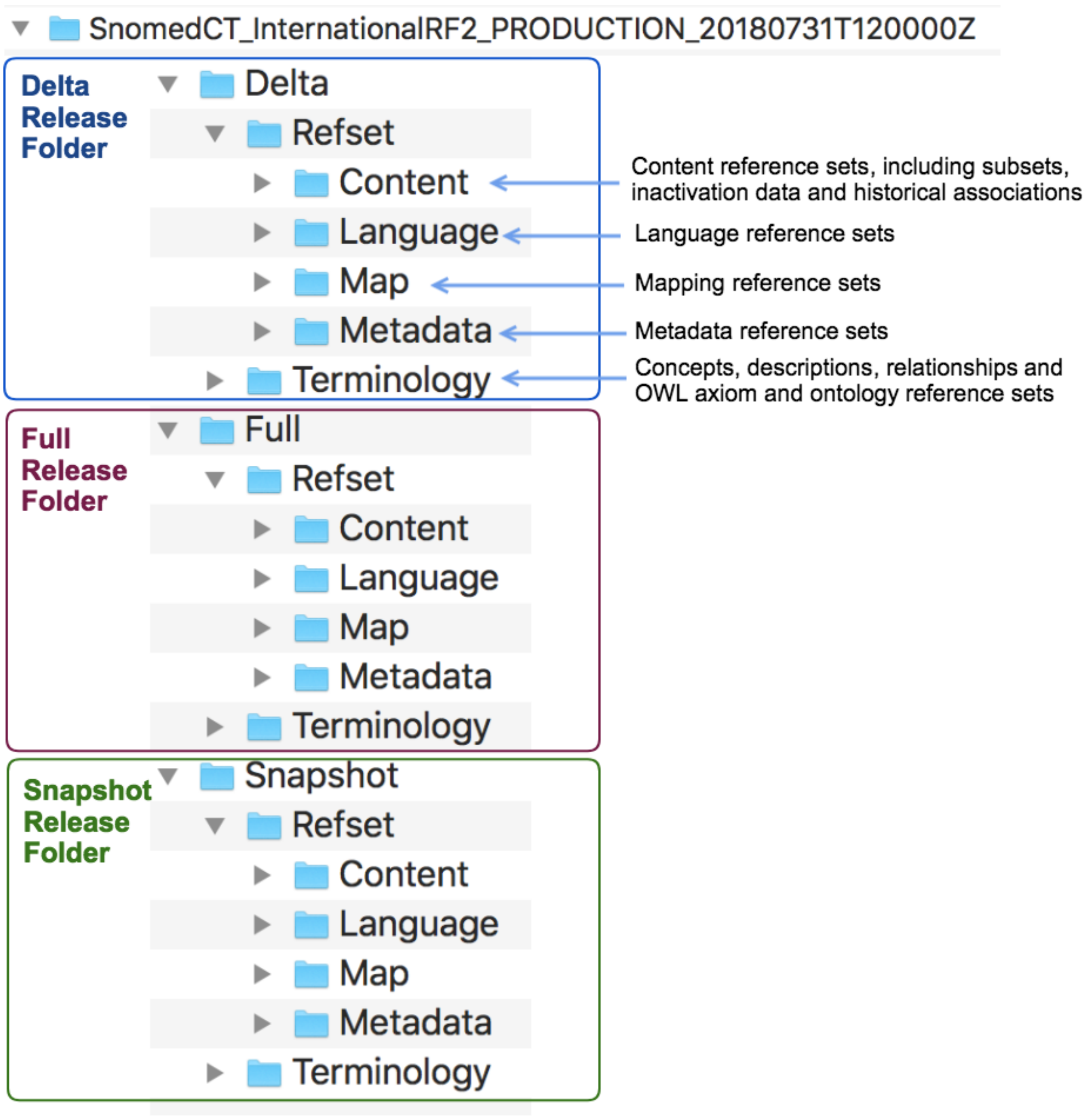
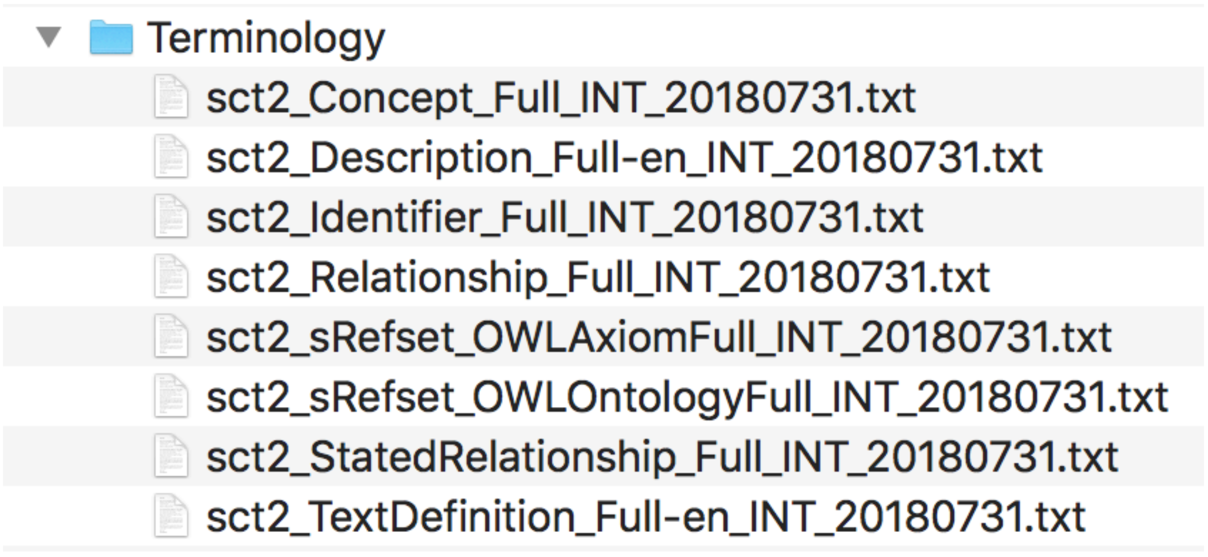


Figure 3.4-1: Release package folder structure

figure 3.4-0 shows the contents of the Terminology folder. In this case, the files shown are those for the Full release type[1] . Details of most of these files are shown in section 4 Component Release Files Specification, while the OWL refsets follow the specification of the OWL Expression Reference Set.

Notes

- The Identifier file contains no data in the International release and can be ignored.
- The Stated Relationship file will be deprecated at the end of a transition period that began in July 2018 and is scheduled to be completed in 2019.

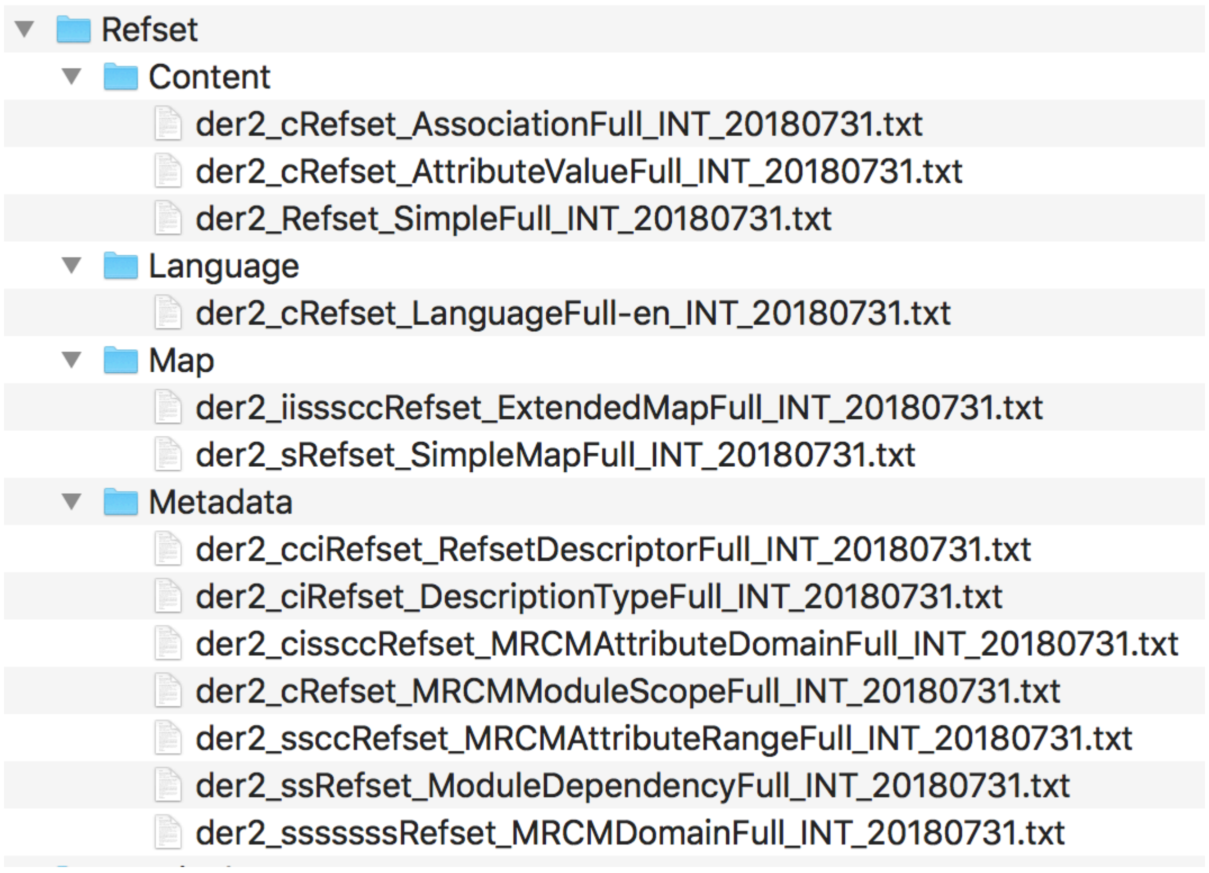


```
▼ Terminology
    sct2_Concept_Full_INT_20180731.txt
    sct2_Description_Full-en_INT_20180731.txt
    sct2_Identifier_Full_INT_20180731.txt
    sct2_Relationship_Full_INT_20180731.txt
    sct2_sRefset_OWLAxiomFull_INT_20180731.txt
    sct2_sRefset_OWLOntologyFull_INT_20180731.txt
    sct2_StatedRelationship_Full_INT_20180731.txt
    sct2_TextDefinition_Full-en_INT_20180731.txt
```

Figure 3.4-2: Files in the Full/Terminology folder

figure 3.4-0 shows the contents of the Refset subfolders. In this case, the files shown are those for the Full release type[2] . The names of the reference set files correspond the reference set types and the structure of each reference set type is specified in section 5.2 Reference Set Types.

Other release packages may include different collections of reference sets to support the intended uses of that package. Similarly, subsequent releases of a package may include additional reference set files that support additional functionality.



- Refset
  - Content
    - der2_cRefset_AssociationFull_INT_20180731.txt
    - der2_cRefset_AttributeValueFull_INT_20180731.txt
    - der2_Refset_SimpleFull_INT_20180731.txt
  - Language
    - der2_cRefset_LanguageFull-en_INT_20180731.txt
  - Map
    - der2_iissscRefset_ExtendedMapFull_INT_20180731.txt
    - der2_sRefset_SimpleMapFull_INT_20180731.txt
  - Metadata
    - der2_cciRefset_RefsetDescriptorFull_INT_20180731.txt
    - der2_ciRefset_DescriptionTypeFull_INT_20180731.txt
    - der2_cissccRefset_MRCMAttributeDomainFull_INT_20180731.txt
    - der2_cRefset_MRCMModuleScopeFull_INT_20180731.txt
    - der2_ssccRefset_MRCMAttributeRangeFull_INT_20180731.txt
    - der2_ssRefset_ModuleDependencyFull_INT_20180731.txt
    - der2_sssssssRefset_MRCMDomainFull_INT_20180731.txt

Figure 3.4-3: Files in the Full/Refset folders

[1] The same file types are present in the other release type Terminology folders. However, the word "Full" in the filenames is replaced by the appropriate release type name ("Snapshot" or "Delta").

[2] The same refset file types are present in the other release type Refset subfolders. However, the word "Full" in the filenames is replaced by the appropriate release type name ("Snapshot" or "Delta").

4 Component Release Files Specification

This guide describes SNOMED CT Release Format 2 (RF2), to be used for official production releases of SNOMED CT. This format is not mandated for internal terminology development usage or as an interchange mechanism between terminology development systems. RF2 provides a format that is flexible, unambiguous and useful. It was designed to strengthen SNOMED CT by providing a simple and stable format that enables innovation through adaptations to cater for changing requirements[1] .

The component release files are defined in the following sections:

---

[1] This specification was developed by harmonizing proposals reviewed by IHTSDO Enhanced Release Format Project Group, including:
- Enhanced Release Format Specification (SNOMED International Proposed Specification , 21 June 2007);
- Reference Set Specification (SNOMED International Proposed Specification , 31 July 2007);
- Alternate Release Format (proposed by NEHTA and their Australian Affiliates).

4.1 Associations Between Release Files

Associations between Component Files

The logical model of associations between the components in the release files is shown in Figure 4.1-1 . The component class represents columns present in all three component files. The individual classes (description, concept and relationship) only show the additional columns present in those files. The colored lines between descriptions and concepts and between relationships and concepts represent the link between the foreign keys (shown in bold) and the id of the concept. These provide the functional connections between components described in this document. The grey lines indicate additional links between columns that are populated with concept identifiers that provide enumerated values.

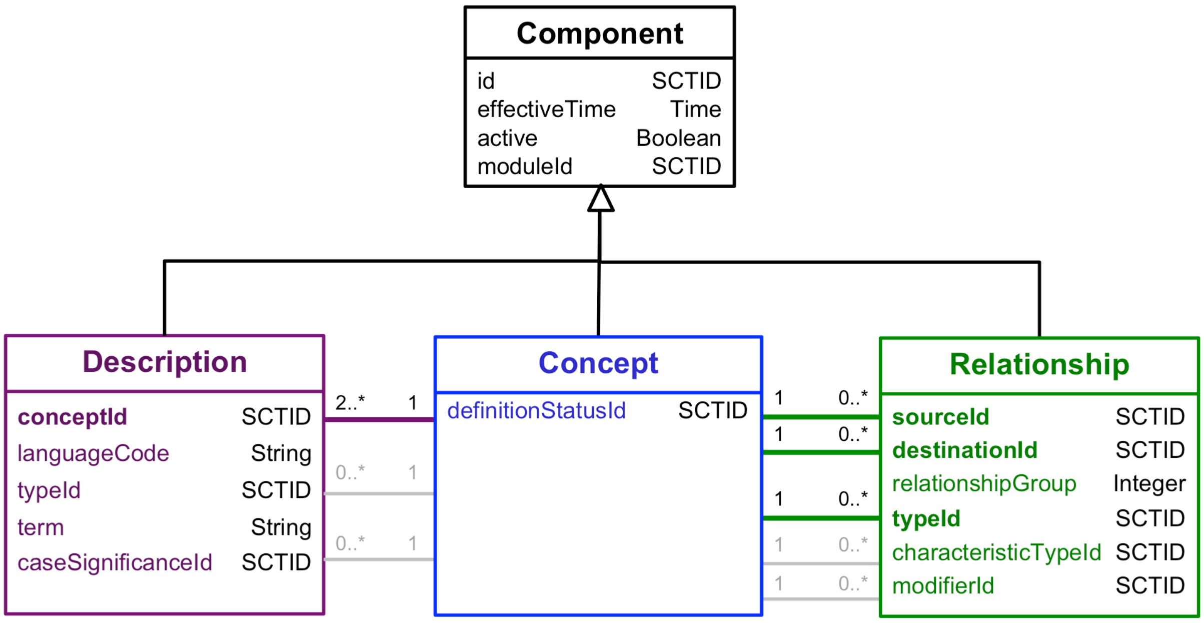


Figure 4.1-1: Logical Relationships Between Component Files

A More Complete View of Release File Associations

Figure 4.1-2 provides an extended view of the associations between release files following changes complete in July 2019 release of SNOMED CT[1] . These changes enable SNOMED CT to use enhanced description logic features and resulted in a significant change to the way in which the stated view of concept definitions are represented. However the changes but did not significantly affect the structure and associations between the main component files shown in Figure 4.1-1

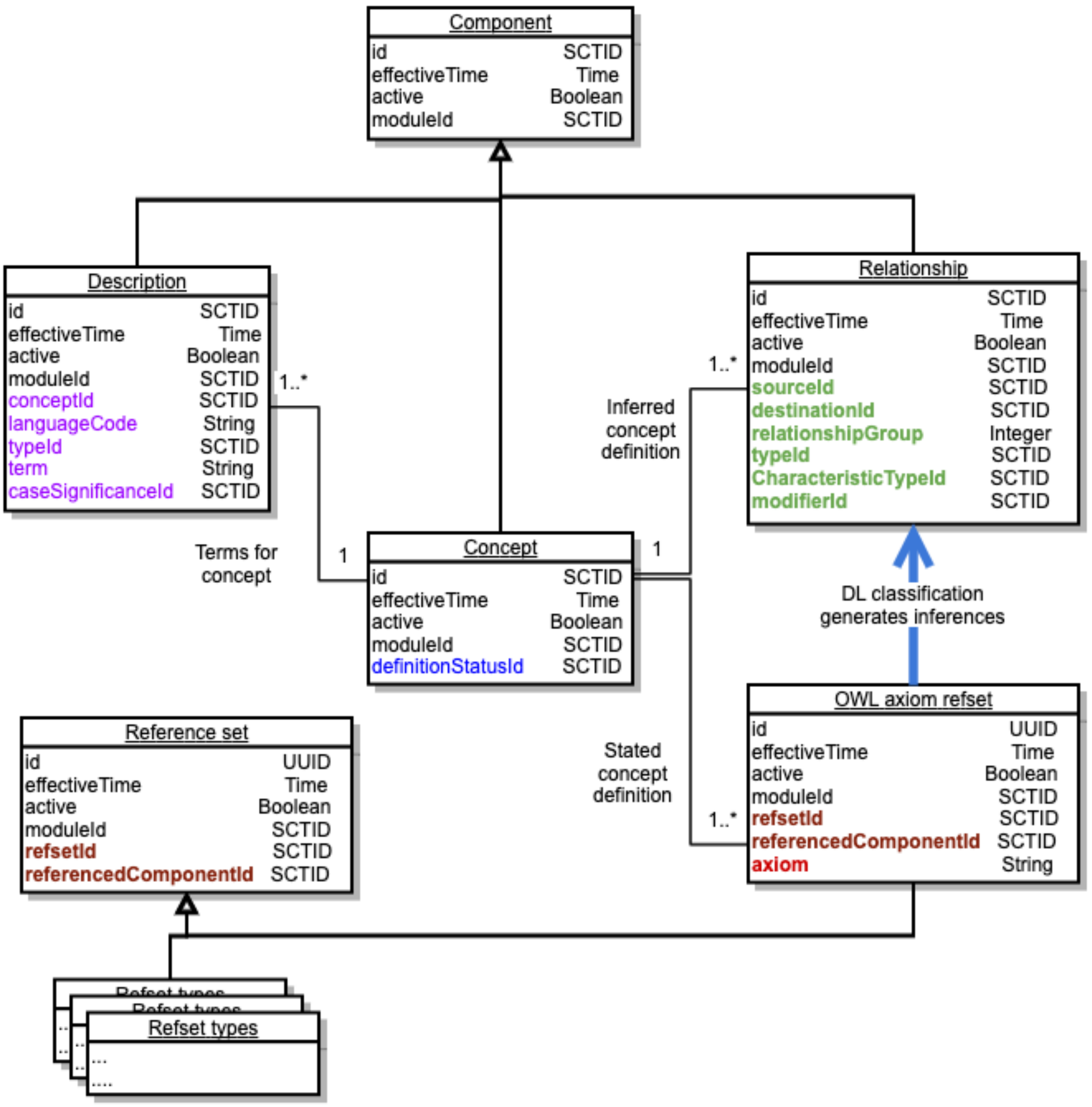


Figure 4.1-2: Associations between SNOMED CT Release Files

Detailed Notes of Release File Associations

Each concept is represented by a row in the Concept and the concept is identified by the id column in that row. There can be more than one row with the same id but with different effectiveTime values, in which case each of these rows represents a version of that same concept. Thus each row represents a version of a clinical concept.

Each concept has two or more descriptions associated with it:

- At least one Fully Specified Name; and
- At least one synonym.

Each description is represented by a row in the description file and is identified by the id column in that row. There can be more than one row with the same id but with different effectiveTime values, in which case each of these rows represents a version of that same description. Thus each row represents a version of a description. Each description applies to one concept to which it is linked by the conceptId. All versions of a description must relate to exactly the same identified concept (i.e. the conceptId must not change between versions).

Each relationship, from a source concept to a destination concept, is represented by a row in the relationship file. There can be more than one row with the same id but with different effectiveTime values, in which case each of these rows represents a version of that same relationship. Thus each row represents a version of a relationship. The source, destination and type each relationship are identified respectively by the sourceId, destinationId and typeId columns. All versions of a relationship must have the same sourceId, destinationId and typeId. The typeId refers to concept, that is also held within the concept file. The only concepts that can be used as the relationship typeId are 116680003 |is a| or concepts that are subtypes of 410662002 |Concept model attribute|.

The most basic form of relationship is the 116680003 |is a| relationship. This relationship states that one concept is a subtype of another concept. Each subtype concept is connected to its parent subtype(s) by relationships with the typeId 116680003 |is a| and this form the main SNOMED CT hierarchy. In this hierarchy, a child concept may have more than one parent concept. The root of the hierarchy is 138875005 |SNOMED CT Concept|, which has a set of top level children, each forming its own sub-hierarchy. Relationships with typeid values that are subtypes of 410662002 |Concept model attribute| are referred to as attribute relationship and contribute to the formal definition of the source concept.

The associations shown on the page are the results of changes that occured between July 2018 and July 2019. For documentation file associations before these changes please refer to Associations Between Release Files Prior to July 2018.

4.2 File Format Specifications

An SQL schema, which represents the content of each of the files specified in the section as a relational table, is provided as part of the Terminology Services Guide (see TSG 1.3 Example of a Full View Relational Representation).

4.2.1 Concept File Specification

The Concept File 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. A concept may be distinguished from or refined by association with other concepts using relationships, which are held in the Relationship.

Table 4.2.1-1: Concept file - Detailed Specification

Field	**Data type**	**Purpose**	Mutable	**Part of Primary Key**
id	SCTID	Uniquely identifies the concept.	NO	YES (Full/ Snapshot)
effectiveTime	Time	Specifies the inclusive date at which the component version's state became the then current valid state of the component. **Note**: In distribution files the effectiveTime should follow the short ISO date format (*YYYYMM DD*) and should not include the hours, minutes, seconds or timezone indicator.	YES	YES (Full) Optional (Snapshot)
active	Boolean	Specifies whether the concept was active or inactive from the nominal release date specified by the effectiveTime.	YES	NO
moduleId	SCTID	Identifies the concept version's module. Set to a descendant of 900000000000443000 | Module | within the metadata hierarchy.	YES	NO
definitionStatusId	SCTID	Specifies if the concept version is primitive or defined. Set to a descendant of 900000000000444006 | Definition status | in the metadata hierarchy.	YES	NO

Only one concept record with the same id field is current at any point in time. The current record will be the one with the most recent effectiveTime before or equal to the date under consideration. If the active field of this record is false ('0'), then the concept is inactive at that point in time.

When a concept is made inactive, the following operations take place:

- A new row is added to the Concepts file for the concept, with the active flag set to inactive and the definitionStatusId set to primitive;
- All relationships that have as source the concept to be inactivated will themselves be inactivated by adding a new row to the Relationship for each relationship, with the active flag set to inactive;
- All active descriptions associated with the concept will remain unchanged unless incorrect for the concept;
- Rows will be added as needed to the Historical Association Reference Sets, to model associations from the inactive concept to other concepts;
- Active descriptions that are still associated with the inactive concept will be added to the 900000000000490003 |Description inactivation indicator reference set|, with an associated value of 900000000000495008 |Concept non-current|

Related Links

- 3.1.1. Concepts
- Appendix C. Unicode UTF-8 encoding
- Concept
- 2.1 High Level Logical Model of SNOMED CT

4.2.2 Description File Specification

The Description holds descriptions that describe SNOMED CT concepts. A description is used to give meaning to a concept and provide well-understood and standard ways of referring to a concept.

Table 4.2.2-1: Description file - Detailed Specification

Field	**Data type**	**Purpose**	Mutable	**Part of Primary Key**
id	SCTID	Uniquely identifies the description.	NO	YES (Full/ Snapshot)
effectiveTime	Time	Specifies the inclusive date at which the component version's state became the then current valid state of the component **Note** : In distribution files the effectiveTime should follow the short ISO date format (*YYYYMM DD*) and should not include the hours, minutes, seconds or timezone indicator.	YES	YES (Full) Optional (Snapshot)
active	Boolean	Specifies whether the state of the description was active or inactive from the nominal release date specified by the effectiveTime .	YES	NO
moduleId	SCTID	Identifies the description version's module. Set to a child of 900000000000443000 | Module | within the metadata hierarchy.	YES	NO
conceptId	SCTID	Identifies the concept to which this description applies. Set to the identifier of a concept in the 138875005 | SNOMED CT Concept | hierarchy within the Concept. Note that a specific version of a description is not directly bound to a specific version of the concept to which it applies. Which version of a description applies to a concept depends on its effectiveTime and the point in time at which it is accessed.	NO	NO
languageCode	String	Specifies the language of the description text using the two character ISO-639-1 code. Note that this specifies a language level only, not a dialect or country code.	NO	NO
typeId	SCTID	Identifies whether the description is fully specified name a synonym or other description type. This field is set to a child of 900000000000446008 | Description type | in the Metadata hierarchy.	NO	NO
term	String	The description version's text value, represented in UTF-8 encoding.	YES	NO
caseSignificanceId	SCTID	Identifies the concept enumeration value that represents the case significance of this description version. For example, the term may be completely case sensitive, case insensitive or initial letter case insensitive. This field will be set to a child of 900000000000447004 | Case significance | within the metadata hierarchy.	YES	NO

Only one description record with the same id field will be current at any point in time. The current record will be the one with the most recent effectiveTime before or equal to the point in time under consideration.

If the active field of this record is false ('0'), then the description is inactive at that point in time. If the active field is true ('1'), then the description is associated with the concept identified by the conceptId field.

The conceptId field, the languageCode field and the typeId field will not change between two rows with the same id, in other words they are immutable. Where a change is required to one of these fields, then the component will be inactivated (by appending a row with the same id and the active field set to false) and a another row will be added representing a new component with a new id. Only limited changes may be made to the term field, as defined by editorial rules.

Each concept will have at least one active description with a typeId of 900000000000013009 |synonym|and at least one active description with a typeId of 900000000000003001 |Fully specified name|.

Where a concept only has one active description with a typeId of 900000000000003001 |Fully specified name|across all language codes within a release, then that Description can be taken as the Fully Specified Name for all languages and dialects, and need not be explicitly included in every language reference set associated with that release.

The term field will be restricted as follows:

- to an overall maximum length of 32Kb;
- to a maximum length, configurable for each description type as defined in the 900000000000538005 |Description format reference set|member associated with that description type- see the Description Format Reference Set specifications document for more details.
- The 900000000000538005 |Description format reference set|also defined the format of the term field (plain text, limited HTML, XHTML) for each description type.
- Control characters (including TABs, CRs and LFs) will not appear in 900000000000540000 |Plain text|or 900000000000541001 |Limited HTML| format types.

Related Links

- 3.1.2. Descriptions and Terms
- 5.2.13 Description Format Reference Set
- Appendix C. Unicode UTF-8 encoding
- Description

4.2.3 Relationship File Specification

The Relationship file holds one relationship per row. Each relationship is of a particular type, and has a source concept and a destination concept. An example of a relationship is given below: 371883000 |Outpatient procedure| 116680003 |Is a| 71388002 |Procedure| where:

- 371883000 |Outpatient procedure| is the source concept;
- 116680003 |Is a| is the relationship type concept and;
- 71388002 |Procedure| is the destination concept.

Table 4.2.3-1: Relationship file - Detailed specification

Field	**Data type**	**Purpose**	Mutable	**Part of Primary Key**
id	SCTID	Uniquely identifies the relationship.	NO	YES (Full/ Snapshot)
effectiveTime	Time	Specifies the inclusive date at which the component version's state became the then current valid state of the component. **Note** : In distribution files the effectiveTime should follow the short ISO date format (*YYYYMM DD*) and should not include the hours, minutes, seconds or timezone indicator.	YES	YES (Full) Optional (Snapshot)
active	Boolea n	Specifies whether the state of the relationship was active or inactive from the nominal release date specified by the effectiveTime field.	YES	NO
moduleId	SCTID	Identifies the relationship version's module. Set to a child of 900000000000443000 | Module | within the metadata hierarchy.	YES	NO
sourceId	SCTID	Identifies the source concept of the relationship version. That is the concept defined by this relationship. Set to the identifier of a concept.	NO	NO
destinationId	SCTID	Identifies the concept that is the destination of the relationship version. That is the concept representing the value of the attribute represented by the typeId column. Set to the identifier of a concept. Note that the values that can be applied to particular attributes are formally defined by the Error rendering macro 'sp-nobody-link' Conversion context did not contain original content entity. .	NO	NO
relationshipGroup	Integer	Groups together relationship versions that are part of a logically associated relationshipGroup. All active Relationship records with the same relationshipGroup number and sourceId are grouped in this way.	YES	NO

typeId	SCTID	Identifies the concept that represent the defining attribute (or relationship type) represented by this relationship version. That is the concept representing the value of the attribute represented by the typeId column. Set to the identifier of a concept. The concept identified must be either 116680003 | Is a | or a subtype of 410662002 | Concept model attribute | . The concepts that can be used as in the typeId column are formally defined as follows: 116680003 | is a | OR < 410662002 | concept model attribute | Note that the attributes that can be applied to particular concepts are formally defined by the Error rendering macro 'sp-nobody-link' Conversion context did not contain original content entity. .	NO	NO
characteristicTypeId	SCTID	A concept enumeration value that identifies the characteristic type of the relationship version (i.e. whether the relationship version is defining, qualifying, etc.) This field is set to a descendant of 900000000000449001 | Characteristic type | in the metadata hierarchy.	YES	NO
modifierId	SCTID	A concept enumeration value that identifies the type of Description Logic (DL) restriction (some, all, etc.). Set to a child of 900000000000450001 | Modifier | in the metadata hierarchy. Currently the only value used in this column is 900000000000451002 | Some | and thus in practical terms this column can be ignored. For further clarification please see **Error rendering macro 'sp-nobody-link'** Conversion context did not contain original content entity. .	YES	NO

Only one relationship record with the same id field will be current at any point in time. The current record will be the one with the most recent effectiveTime before or equal to the point in time under consideration.

If the active field of this record is false ('0'), then the relationship is inactive at that point in time. If the active field is true ('1'), then there is a relationship between the SNOMED CT concepts identified by sourceId and destinationId.

The sourceId, destinationId, relationshipGroup, typeId, characteristicTypeIdand modifierId will not change between two rows with the same id, in other words they are immutable. Where a change is required to one of these fields, then the current row will be de-activated (by appending a row with the same id and the active field set to false) and a new row with a new id will be appended.

The relationshipGroup field is used to group relationships with the same sourceId field into one or more logical sets. A relationship with a relationshipGroup field value of '0' is considered not to be grouped. All relationships with the same sourceId and non-zero relationshipGroup are considered to be logically grouped.

The relationshipGroup field will be an unsigned Integer, and will not be limited to a single digit value. There is no guarantee that they will be assigned sequentially, and the values will not be unique across concepts.

Related Links

- SNOMED CT Machine Readable Concept Model.
- 3.1.3. Relationships
- Appendix C. Unicode UTF-8 encoding
- Relationship

4.2.4 Identifier File Specification

Important Note
The Identifier File does not contain any data in the SNOMED CT International Release
The file structure is documented here only as a point of reference for others who may be using the files in an extension release.

This file provides a standardized way of associating alternative Identifiers from various schemes with SNOMED CT components.

At any point in time , an alternative Identifier within a particular scheme will be associated with one and only one SNOMED CT component. A SNOMED CT component may be associated with zero or more alternative Identifiers within a single scheme.

It is important to note that the SNOMED CT component and it's alternative Identifiers all identify precisely the same real-world object.

Note: The Identifier file is not currently used in the SNOMED CT International Release as use of the more flexible Simple map type references set structure is preferred for links to alternative codes. The only known current use of this file is for internal identification of components during the content development process.

Table 4.2.4-1: Identifier file - Detailed Specification

Field	**Data type**	**Purpose**	Mutable	**Part of Primary Key**
identifierSchemeId	SCTID	Identifier of the concept enumeration value from the Metadata hierarchy that represents the scheme to which the Identifier value belongs. Set to a descendant of 900000000000453004 | Identifier scheme | within the metadata hierarchy.	NO	YES (Full/ Snapshot)
alternateIdentifier	String	String representation of the alternative Identifier in its native scheme.	NO	YES (Full/ Snapshot)
effectiveTime	Time	Specifies the inclusive date at which the alternative Identifier was associated with the SNOMED CT component.	YES	YES (Full) Optional (Snapshot)
active	Boolean	Specifies whether the association was active or inactive from the point in time specified by the effectiveTime .	YES	NO
moduleId	SCTID	Identifies the source module that this association was created in. Set to a child of 900000000000443000 | Module | within the metadata hierarchy.	YES	NO
referencedComponentId	SCTID	Uniquely identifies the SNOMED CT component with which the alternative Identifier is associated.	NO	NO

Only one record with the same identifierSchemeId and alternateIdentifier fields will be current at any point in time. The current record will be the one with the most recent effectiveTime before or equal to the point in time under consideration.

If the active field of this record is false ('0'), then the association is inactive at that point in time. If the active field is true ('1'), then there is an identity at that point in time between the referencedComponentId (a SNOMED CT component) and the alternateIdentifier in the scheme identified by identifierSchemeId.

4.2.5 Transitive Closure Files

Important Note
Transitive Closure Files are not distributed in the SNOMED CT International Release
The file structures documented here are points of reference for those generating transitive closure table from release data. SNOMED International provides a script file that can be used to generate the Snapshot Transitive Closure file from the snapshot Relationship File.

The Transitive Closure is the complete set of relationships between every concept and each of its super-type concepts, in other words both its parents and ancestors.

A transitive closure table is one of the most efficient ways to test for subsumption between concepts.

Snapshot Transitive Closure File

SNOMED International provides an example of a Transitive Closure Perl script file (click to download) that can be used to generating a snapshot view of the transitive closure from the snapshot release of the Relationship. The output of this script conforms to the following following file structure. Note that the primary key for this table consists of both columns.

Table 4.2.5-2: Transitive Closure File - Detailed Specification

Field	**Data type**	**Purpose**	**Part of Primary Key**
subtypeId	SCTID	Id of the concept playing the subtype role. Set to an Identifier of a concept.	YES
supertypeId	SCTID	Id of the concept playing the supertype role. Set to an Identifier of a concept.	YES

Versioned Transitive Closure

A versioned view of the Transitive Closure can also be generated by combining the snapshot views for different effective times and removing redundant rows (e.g. where the transitive closure has not changed between release versions). The generated file could then be represented using the example specification below. Note that the unique key for this file would consist of the **subtypeId**, **supertypeId** and **effectiveTime**.

Table 4.2.5-2: Versioned Transitive Closure File - Example Specification

Field	Data type	Purpose	Part of Primary Key
subtypeId	SCTID	Id of the concept playing the subtype role. Set to an Identifier of a concept.	YES
supertypeId	SCTID	Id of the concept playing the supertype role. Set to an Identifier of a concept.	YES
effectiveTime	Time	Specifies the inclusive date at which the transitive closure record was added or changed its active state.	YES
active	Boolean	Specifies whether at the transitive closure represented by the subtypeId and supertypeId became valid (active) or invalid (inactive) from the point in time specified by the effectiveTime.	NO

Related Links

- 7.5.2 Transitive closure implementation

4.3 Metadata Hierarchy

As the release file formats contain a number of concept enumerations, it is necessary to define sets of concepts that represent the allowed values. As well as the enumerated values, other metadata supporting the extensibility mechanism and the concept model is required.

The concept 900000000000441003 |SNOMED CT Model Component (metadata)|is a subtype of the root concept(138875005 |SNOMED CT Concept|), and contains the metadata, supporting the release.

The subtypes of 900000000000441003 |SNOMED CT Model Component (metadata)|are described in the following table and the top three levels of the hierarchy are shown in the figure below this.

Table 4.3-3: SNOMED CT Model Component (metadata) (900000000000441003)

Id	Term	Comment
106237007 |Linkage concept (linkage concept)|	106237007 |Linkage concept (linkage concept)|	Concepts that specify • Semantic Relationships between concepts(246061005 |Attribute|); and • Asserted associations between statements in a record (416698001 |Link assertion|)
370136006 |Namespace concept (namespace concept)|	370136006 |Namespace concept (namespace concept)|	Concepts that specify the Extension Namespaces allocated by the SNOMED International.
900000000000442005 |Core metadata concept (core metadata concept)|	900000000000442005 |Core metadata concept (core metadata concept)|	Concepts that are referenced from enumerated fields within the International Release files (the Concept, Description, Relationship, Identifier).
900000000000454005 |Foundation metadata concept (foundation metadata concept)|	900000000000454005 |Foundation metadata concept (foundation metadata concept)|	The metadata that supports the extensibility mechanism, and is discussed in more detail in the Reference Sets Guide.

Table 4.3-3: SNOMED CT Metadata Hierarchy (2018-01-31) - Core metadata concepts (top 3 levels only)

- 900000000000441003 |SNOMED CT Model Component|
  - 900000000000442005 |Core metadata concept|
    - 900000000000447004 |Case significance|
      - 900000000000448009 |Case insensitive|
      - 900000000000017005 |Case sensitive|
      - 900000000000020002 |Initial character case insensitive|
    - 900000000000449001 |Characteristic type|
      - 900000000000227009 |Additional relationship|
      - 900000000000006009 |Defining relationship|
      - 900000000000225001 |Qualifying relationship|
    - 900000000000444006 |Definition status|
      - 900000000000073002 |Defined|
      - 900000000000074008 |Primitive|
    - 900000000000446008 |Description type|
      - 900000000000550004 |Definition|
      - 900000000000003001 |Fully specified name|
      - 900000000000013009 |Synonym|
    - 900000000000453004 |Identifier scheme|
      - 900000000000294009 |SNOMED CT integer ID|
      - 900000000000002006 |SNOMED CT UUID|
    - 900000000000450001 |Modifier|
      - 900000000000452009 |All|
      - 900000000000451002 |Some|
    - 900000000000443000 |Module|
      - 900000000000445007 |IHTSDO maintained module|
      - 466707005 |SNOMED CT Medical Devices module|
  - 900000000000454005 |Foundation metadata concept| ... (see next table)

Table 4.3-3: SNOMED CT Metadata Hierarchy (2018-01-31) - Foundation metadata concepts (top 3 levels only - some long lists replaced by ...)

```
900000000000441003 |SNOMED CT Model Component|
  900000000000442005 |Core metadata concept| ... (see previous table)
  900000000000454005 |Foundation metadata concept|
    900000000000455006 |Reference set|
      900000000000516008 |Annotation type|
      900000000000521006 |Association type|
      900000000000480006 |Attribute value type|
      705109006 |Code to expression type reference set|
      447250001 |Complex map type reference set|
      609430003 |Concept model reference set|
      900000000000538005 |Description format|
      733614007 |Expansion history reference set|
      609331003 |Extended map type reference set|
      733613001 |Intensional definition reference set|
      900000000000506000 |Language type|
      705111002 |Map correlation and origin type reference set|
      900000000000534007 |Module dependency|
      723564002 |MRCM reference set|
      733618005 |Ordered association type reference set|
      733619002 |Ordered component type reference set|
      447258008 |Ordered type reference set|
      900000000000512005 |Query specification type|
      900000000000456007 |Reference set descriptor|
      900000000000496009 |Simple map|
      446609009 |Simple type reference set|
    900000000000457003 |Reference set attribute|
      447257003 |"Linked to" reference set attribute|
      900000000000511003 |Acceptability|
      ...
      723569007 |Template|
  106237007 |Linkage concept|
    246061005 |Attribute|
      410663007 |Concept history attribute|
      410662002 |Concept model attribute|
      116680003 |Is a|
      408739003 |Unapproved attribute|
    416698001 |Link assertion|
      417151001 |Has explanation|
      ...
      416872009 |Is etiology for|
      417318003 |Is manifestation of|
  370136006 |Namespace concept|
    373872000 |Core Namespace|
    370137002 |Extension Namespace 1000000|
    ...
    713754005 |Extension Namespace 1000999|
```

5 Reference Set Release Files Specification

This section of the SNOMED CT Release Files Specification provides details of the structure and content of reference set files distributed by SNOMED International as part of the SNOMED CT International Release. This is also the standard format in which producers of SNOMED CT extension are required to distribute any reference sets that they produce to their sublicensees[1].

Reference set data structures provide a generic mechanism for configuration and extensibility of SNOMED CT to a wide range of different requirements. Reference sets act as building blocks that provide a common foundation that enables those developing SNOMED CT extensions to customize the way their users interact with SNOMED CT. The flexibility offered by reference sets also enables adaptation of existing system to support changing requirements.

The reference set file formats are formally defined in following subsections.

---

[1] The files specified by this section form part of SNOMED CT Release Format 2 (RF2) - the standard release format for SNOMED CT since 2012. RF2 is a flexible, simple stable format with support for robust versioning. It enables innovation through adaptations to cater for changing requirements. This format is only mandated as the standard distribution format for SNOMED CT International and SNOMED CT Extensions. Internal representations of SNOMED CT resources within an application may vary provided such representations faithfully retain the information represented in the release files.

5.1 General Features of Reference Sets

This section provides summary information on the general features of reference sets. Section 5.2 Reference Set Types build on this providing detailed specifications of each of the internationally defined reference sets. However, for more detail of the purposes for which reference sets can be used and the ways in which the design of different reference set types meet practical requirements, please see the Practical Guide to Reference Sets.

5.1.1 Basic Reference Set Member File Format

The basic reference set data structure consists of the following fields:

Table 5.1.1-1: Basic Reference Set Data Structure

Field	**Data type**	**Purpose**	Mutable	**Part of Primary Key**
id	UUID	A 128 bit unsigned Integer, uniquely identifying this reference set member. Different versions of a *reference set member* share the same id but have different effectiveTime. This allows a *reference set member* to be modified or made inactive (i.e. removed from the active set) at a specified time.	NO	YES (Full / Snapshot)
effectiveTime	Time	The inclusive date or time at which this version of the identified reference set member became the current version. **Note** : In distribution files the effectiveTime should follow the short ISO date format (*YYYYMM DD*) and should not include the hours, minutes, seconds or timezone indicator. The current version of this reference set member at time *T* is the version with the most recent effectiveTime prior to or equal to time *T* .	YES	YES (Full) Optional (Snapshot)
active	Boolean	The state of the identified reference set member as at the specified effectiveTime . If active = 1 (true) the reference set member is part of the current version of the set, if active = 0 (false) the reference set member is not part of the current version of the set.	YES	NO

moduleId	SCTID	Identifies the SNOMED CT module that contains this reference set member as at the specified effectiveTime . The value must be a subtype of 900000000000443000 | Module (core metadata concept) | within the metadata hierarchy .	YES	NO
refsetId	SCTID	Identifies the reference set to which this reference set member belongs.	NO	NO
referencedComponentId	SCTID	A reference to the SNOMED CT component to be included in the reference set .	NO	NO
Zero or more other fields dependent on reference set type	SCTID , String , or Integer	Optional field(s) serving purposes specific to the reference set type. For details see Error rendering macro 'sp-nobody-link' Conversion context did not contain original content entity. .	YES	NO

Each reference set is identified and named by a concept in the metadata hierarchy. Therefore the reference set is identified by a concept identifier (an SCTID).

Each row in a reference set file represents a reference set member.

- Individual reference set members are uniquely identified by a identifier represented as a UUID.
- Each reference set member belongs to a single reference set, and it is linked to that reference set by the refsetId field.
- Each reference set member is also associated with a single referenced component by its referencedComponentId field. The referenced component may be a concept, description, relationship. If the referenced component is a concept that identifies another reference set, that reference set may be considered to be the target of the reference.
- Like components, reference set members can be versioned to inactivate or change the status of the member. So there may be several rows in a full release file and in this case the one with the most recent effectiveTime before or equal to the point in time under consideration represents state of that reference set member. If the active field of this row is false ('0'), then the reference set member is inactive at that point in time, which means that component it refers to is not a member of the reference set. If the active field is true ('1'), then the component referenced by the referencedComponentId field is deemed to be a member of the reference set.

The refsetId and referencedComponentId fields will not change between two rows with the same id, in other words they are immutable. Where a change is required to one of these fields, the current row will be inactivated (by appending a row with the same id and the active field set to false). Another row with a new id will be appended to reference another component.

A component may belong to any number of reference sets. A component may also be referenced by more that one member of the same reference set. This is not useful in the case of a simple reference set but is relevant for some reference sets. For example, a SNOMED CT concept may map to or from more than on codes in another code system.

4.1.2. Extending the Basic Reference Set Member File Format

The basic reference set file structure enables representation of subsets fo SNOMED CT components. However, the reference set format is extensible, allowing it to be used for a wide range of other purposes. For more details about the requirements that reference sets can address please see the Practical Guide to Reference Sets.

The basic reference set structure can be extended by adding one or more fields. Each of these fields will hold additional specific values related to each member. Three general data types are supported in the additional columns. These are

- Integer
- String and
- Component (a reference to a SNOMED CT component)

Finer grained interpretation of these data types can also be specified using a special metadata reference set known as the 900000000000456007 |Reference set descriptor|.

The reference Set patterns that are supported as part of the International Edition are documented in 5.2 Reference Set Types. Additional reference set patterns can also be created as part of an extension to support additional use case (see 4.1.2. Extending the Basic Reference Set Member File Format).

5.1.3 Naming Conventions for Reference Sets

National Release Centres and others may create additional reference sets. A namespace is required to create a new reference set, as each reference set is defined by a concept. The concept's FSN and a synonym are used to name the reference set. Where a new reference set is created against an existing pattern, then the following naming convention should be used (where the text "*Specific name*" is to be replaced by the specific name of the reference set).

Attribute Value Reference Set

- FSN = *Specific name* attribute value reference set (foundation metadata concept)
- PT = *Specific name* attribute value map

Language Reference Set

For a Language:

- FSN = *Language name* [International Organization for Standardization 639-1 code *Language code*] language reference set (foundation metadata concept)
    - Example: 900000000000507009 English [International Organization for Standardization 639-
- PT = *Language name*
    - Example: 900000000000507009 |English|

For a Dialect:

- FSN = *Dialect name* *Language name* language reference set (foundation metadata concept)
    - Example:
- PT = *Dialect code* *Language name*
    - Example: 900000000000508004 |GB English|

Annotation Reference Set

- FSN = *Specific name* annotation reference set (foundation metadata concept)
- PT = *Specific name* annotation reference set

Association Reference Set

- FSN = *Specific name* association reference set (foundation metadata concept)
- PT = *Specific name* association reference set

5.1.4 Metadata Supporting Reference Sets

Reference sets types are identified by concepts that are subtypes of the metadata concept 900000000000455006 |reference set|. Individual reference sets of a particular type are identified and named by concepts that are subtype descendants of the concept that identifies the reference set type.

Table 5.1.4-2: Reference Set Types in the Metadata Hierarchy (2018-01-31)

- 900000000000455006 |Reference set|
  - 900000000000516008 |Annotation type|
  - 900000000000521006 |Association type|
  - 900000000000480006 |Attribute value type|
  - 705109006 |Code to expression type reference set|
  - 447250001 |Complex map type reference set|
  - 609430003 |Concept model reference set|
  - 900000000000538005 |Description format|
  - 733614007 |Expansion history reference set|
  - 609331003 |Extended map type reference set|
  - 733613001 |Intensional definition reference set|
  - 900000000000506000 |Language type|
  - 705111002 |Map correlation and origin type reference set|
  - 900000000000534007 |Module dependency|
  - 723564002 |MRCM reference set|
  - 733618005 |Ordered association type reference set|
  - 733619002 |Ordered component type reference set|
  - 447258008 |Ordered type reference set|
  - 900000000000512005 |Query specification type|
  - 900000000000456007 |Reference set descriptor|
  - 900000000000496009 |Simple map|
  - 446609009 |Simple type reference set|

Other concepts within the metadata hierarchy are used to name additional attributes within particular types of reference sets and to provide values for those attributes.

Table 5.1.4-2: Reference Set Attributes in Metadata Hierarchy (2018-01-31) (some omitted)

```
900000000000457003 |Reference set attribute|
  447257003 |"Linked to" reference set attribute|
  900000000000511003 |Acceptability|
  900000000000518009 |Annotated component|
  900000000000519001 |Annotation|
  900000000000532006 |Association source component|
  900000000000533001 |Association target component|
  900000000000458008 |Attribute description|
  900000000000479008 |Attribute order|
  723576002 |Attribute rule|
  900000000000459000 |Attribute type|
  900000000000491004 |Attribute value|
  733616009 |Authoring substrate|
  723571007 |Cardinality|
  609431004 |Concept model domain|
  609432006 |Concept model range|
  609642003 |Concept model relationship type|
  723573005 |Concept model rule strength|
  723574004 |Content type|
  900000000000535008 |Dependency target|
  900000000000539002 |Description format|
  900000000000510002 |Description in dialect|
  900000000000544009 |Description length|
  723565001 |Domain constraint|
  733612006 |Expansion substrate|
  706999006 |Expression|
  900000000000514006 |Generated reference set|
  723572000 |Grouped|
  723570008 |Guide URL|
  900000000000504002 |Map advice|
  609330002 |Map category value|
  900000000000501005 |Map group|
  900000000000502003 |Map priority|
  900000000000503008 |Map rule|
  900000000000500006 |Map source concept|
  900000000000505001 |Map target|
  723577006 |MRCM rule reference set|
  705116007 |Original code system source for linked content value|
  723566000 |Parent domain|
  447255006 |Priority order reference set attribute|
  723567009 |Proximal primitive constraint|
  723568004 |Proximal primitive refinement|
  900000000000515007 |Query|
  733615008 |Query language|
  733617000 |Query string|
  723575003 |Range constraint|
  ... more attributes ...
```

5.2 Reference Set Types

This section describes a number of standard reference set types.

Each reference set type follows a pattern and that pattern is also represented in a machine readable form using a set of Reference Set Descriptor members (known as a Descriptor Template, for short). In most case, the same pattern may be used to define a number of different reference sets to serve a variety of purposes. However, there are also some highly specific reference set types that exist for a single specified purpose. These are the Reference Set Descriptor Reference Set, Module Dependency Reference Set and Description Format Reference Set.

In each subsection, a reference set type is described under the following subheadings:

- The purpose of the reference set;
- The format of the reference set member record is detailed in a table;
- The metadata supporting the reference set;
- The machine readable reference set descriptor member records for the reference set type;
- Examples of the reference set type;

Related Links

- Reference Sets
- Unicode UTF-8 encoding
- Reference set

5.2.1 Simple Reference Set

Purpose

A 446609009 |Simple type reference set| allows a set of components to be specified for inclusion or exclusion for a specified purpose. This type of reference ret represents an extensional definition of a subset of SNOMED CT components. Thus it can be used to fully enumerate a subset of concepts, descriptions or relationships.

See also Query specification reference set, which can be used to represent an intensional definition of a subset of SNOMED CT components. In an intensional definition, the members of the subset are specified by rules rather than by enumerations (e.g. all subtypes of a specified concepts).

Reference Set Data Structure

A Simple reference set does not have any additional fields.

Table 5.2.1-4: Simple Reference Set - Data Structure

Field	**Data type**	**Purpose**	Mutable	**Part of Primary Key**
id	UUID	A 128 bit unsigned Integer, uniquely identifying this reference set member. Different versions of a *reference set member* share the same id but have different effectiveTime. This allows a *reference set member* to be modified or made inactive (i.e. removed from the active set) at a specified time.	NO	YES (Full / Snapshot)
effectiveTime	Time	The inclusive date or time at which this version of the identified reference set member became the current version. **Note**: In distribution files the effectiveTime should follow the short ISO date format (*YYYYMMDD*) and should not include the hours, minutes, seconds or timezone indicator. The current version of this reference set member at time *T* is the version with the most recent effectiveTime prior to or equal to time *T*.	YES	YES (Full) Optional (Snapshot)

active	Boolean	The state of the identified reference set member as at the specified effectiveTime . If active = 1 (true) the reference set member is part of the current version of the set, if active = 0 (false) the reference set member is not part of the current version of the set.	YES	NO
moduleId	SCTID	Identifies the SNOMED CT module that contains this reference set member as at the specified effectiveTime . The value must be a subtype of 900000000000443000 | Module (core metadata concept) | within the metadata hierarchy.	YES	NO
refsetId	SCTID	Identifies the reference set to which this reference set member belongs. A subtype descendant of: • 446609009 | Simple type reference set |	NO	NO
referencedComponentId	SCTID	A reference to the SNOMED CT component to be included in the reference set.	NO	NO

Metadata

Simple References Sets are subtypes of 446609009 |Simple type reference set|in the metadata hierarchy.

Table 5.2.1-4: Simple Reference Sets in the Metadata Hierarchy

- 900000000000441003 |SNOMED CT Model Component|
  - 900000000000454005 |Foundation metadata concept|
    - 900000000000455006 |Reference set|
      - 446609009 |Simple type reference set|

Reference Set Descriptor and Example Data

Notes on the tables used to show descriptors and examples
The reference set example tables on this page have been revised as follows to aid clarity and understanding:

- The first four columns which are present in all release files are not shown. The omitted columns (id, effectiveTime, active, moduleId) are used in the same way in all referenced sets to support identification, versioning and packaging. They do not directly affect the specific features of a particular reference set or reference set type.
- Reference set columns that contain SNOMED CT identifiers are expanded to show details of the concept or description referenced by that identifier. In some cases, the term is shown in the same column using the expression syntax, in other cases an additional column with a name suffix '_term' has been added. In the standard reference set files only the identifier is present in the column and there is no added column for the term. When using reference sets, the term and other details of the component are looked up from the relevant component release files.

Descriptor Template

The table below shows the descriptor for a specific reference sets that follows the 446609009 |Simple type reference set|pattern.

Table 5.2.1-4: Refset Descriptor rows for the Simple Reference Set Type

refsetId	referencedComponentId (Referenced component)	attributeDescription (Attribute description)	attributeType (Attribute type)	attributeOrder (Attribute order)
900000000000456007 |Reference set descriptor|	447566000 |Virtual medicinal product simple reference set|	449608002 |Referenced component|	900000000000461009 |Concept type component|	0

Example Data

Table 5.2.1-4: Example Data for a Simple Reference Set

refsetId	referencedComponentId (Referenced component)
447565001 |Virtual therapeutic moiety simple reference set|	211009 |Norethandrolone preparation|
447565001 |Virtual therapeutic moiety simple reference set|	302007 |Spiramycin|
447565001 |Virtual therapeutic moiety simple reference set|	449005 |Penicillin G procaine|
447565001 |Virtual therapeutic moiety simple reference set|	544002 |Melphalan|
447565001 |Virtual therapeutic moiety simple reference set|	669007 |Vaccinia virus vaccine|
447565001 |Virtual therapeutic moiety simple reference set|	796001 |Digoxin|
447565001 |Virtual therapeutic moiety simple reference set|	847003 |D-thyroxine preparation|
447565001 |Virtual therapeutic moiety simple reference set|	922004 |Pralidoxime|
447565001 |Virtual therapeutic moiety simple reference set|	1039008 |Mercaptopurine|
447565001 |Virtual therapeutic moiety simple reference set|	1148001 |Ticarcillin|

5.2.2 Ordered Component Reference Set

Purpose

An 733619002 |Ordered component type reference set (foundation metadata concept)|allows a collection of components to be defined with a specified order. This type of reference ret is therefore useful for creating ordered lists and to specify groups where the components that belong to the same group share the same order.

Data structure

An Ordered component reference set is an Integer Component reference set is used to represent ordered lists and alternative hierarchies. Its structure is shown in the following table.

Table 5.2.2-4: Ordered component reference set - Data structure

Field	Data type	Purpose	Mutable	Part of Primary Key

id	UUID	A 128 bit unsigned Integer, uniquely identifying this reference set member. Different versions of a *reference set member* share the same id but have different effectiveTime. This allows a *reference set member* to be modified or made inactive (i.e. removed from the active set) at a specified time.	NO	YES (Full / Snapshot)
effectiveTime	Time	The inclusive date or time at which this version of the identified reference set member became the current version. **Note** : In distribution files the effectiveTime should follow the short ISO date format (*YYYYMMDD*) and should not include the hours, minutes, seconds or timezone indicator. The current version of this reference set member at time *T* is the version with the most recent effectiveTime prior to or equal to time *T* .	YES	YES (Full) Optional (Snapshot)
active	Boolean	The state of the identified reference set member as at the specified effectiveTime . If active = 1 (true) the reference set member is part of the current version of the set, if active = 0 (false) the reference set member is not part of the current version of the set.	YES	NO
moduleId	SCTID	Identifies the SNOMED CT module that contains this reference set member as at the specified effectiveTime . The value must be a subtype of 900000000000443000 |Module (core metadata concept)| within the metadata hierarchy.	YES	NO
refsetId	SCTID	Identifies the reference set to which this reference set member belongs. In this case, a subtype descendant of: 900000000000443000 |Module (core metadata concept)|	NO	NO
referencedComponentId	SCTID	A reference to the SNOMED CT component to be included in the reference set.	NO	NO
order	Integer	Specifies the sort order of the list. The list is ordered by applying an ascending sort of the order value. The value of order =1 represents the highest priority. A value of '0' is not allowed. Duplicate values are permitted and the sort order between two members with the same order value is not defined. Note: The name "order" is a reserved word in some database environments. Please consider this when using this column.	YES	NO

Metadata

The following metadata in the "Foundation metadata concept" hierarchy supports this reference set:

Table 5.2.2-4: Ordered Component Reference Sets in the Metadata Hierarchy

```
900000000000454005 |Foundation metadata concept|
    900000000000455006 |Reference set|
        733619002 |Ordered component type reference set|
```

Reference Set Descriptor and Example Data

Notes on the tables used to show descriptors and examples
The reference set example tables on this page have been revised as follows to aid clarity and understanding:

- The first four columns which are present in all release files are not shown. The omitted columns (id, effectiveTime, active, moduleId) are used in the same way in all referenced sets to support identification, versioning and packaging. They do not directly affect the specific features of a particular reference set or reference set type.
- Reference set columns that contain SNOMED CT identifiers are expanded to show details of the concept or description referenced by that identifier. In some cases, the term is shown in the same column using the expression syntax, in other cases an additional column with a name suffix '_term' has been added. In the standard reference set files only the identifier is present in the column and there is no added column for the term. When using reference sets, the term and other details of the component are looked up from the relevant component release files.

Descriptor Template

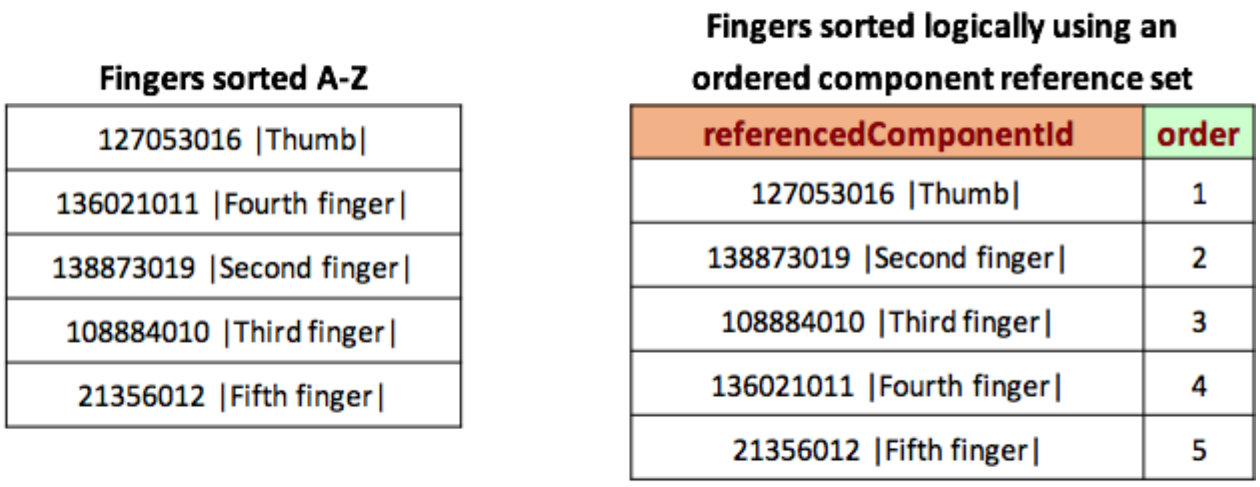
The tables below show the descriptor that defines the structure of the 733619002 |Ordered component type reference set| pattern and an example of descriptor for a specific reference set that follows this pattern.

Table 5.2.2-4: Refset Descriptor rows for an ordered reference set

refsetId	referencedComponentId	attributeDescription	attributeType	attributeOrder
900000000000456007 |Reference set descriptor|	733619002 |Ordered component type reference set|	449608002 |Referenced component|	900000000000460005 |Component type|	0
900000000000456007 |Reference set descriptor|	733619002 |Ordered component type reference set|	447255006 |Priority order reference set attribute|	900000000000478000 |Unsigned integer|	1

Note: The table above omits the initial four columns of data present in the release file. These follow the standards versioning pattern id, effectiveTime, active, active. Additionally, to aid understanding, the table above also shows the term from one of the descriptions associated with each of the identified concept. The release file only contains the identifier.

Ordered reference set example



Fingers sorted A-Z

127053016 |Thumb|
136021011 |Fourth finger|
138873019 |Second finger|
108884010 |Third finger|
21356012 |Fifth finger|

Fingers sorted logically using an ordered component reference set

referencedComponentId	order
127053016 |Thumb|	1
138873019 |Second finger|	2
108884010 |Third finger|	3
136021011 |Fourth finger|	4
21356012 |Fifth finger|	5

Table 5.2.2-4: Rational ordering of finger concepts using an ordered component reference set

refsetId	referencedComponentId (Referenced component)	order (Attribute order)
733619002 | Fingers ordered component reference set |	127053016 |Thumb|	1
733619002 | Fingers ordered component reference set |	138873019 |Second finger|	2
733619002 | Fingers ordered component reference set |	108884010 |Third finger|	3
733619002 | Fingers ordered component reference set |	136021011 |Fourth finger|	4
733619002 | Fingers ordered component reference set |	21356012 |Fifth finger|	5

5.2.3 Attribute Value Reference Set

Purpose

An 900000000000480006 |Attribute value type reference set| allows a value from a specified range to be associated with a component. This type of reference set can be used for a range of purposes where there is a requirement to provide additional information about particular concepts, descriptions or relationships. For example, an 900000000000480006 |Attribute value type reference set| is used to indicate the reason why a concept has been inactivated.

Data Structure

An Attribute value reference set is a component reference set used to apply a tagged value to a SNOMED CT component. Its structure is shown in the following table.

Table 5.2.3-5: Attribute Value Reference Set - Data Structure

Field	Data type	Purpose	Mutable	Part of Primary Key
id	UUID	A 128 bit unsigned Integer, uniquely identifying this reference set member. Different versions of a *reference set member* share the same id but have different effectiveTime. This allows a *reference set member* to be modified or made inactive (i.e. removed from the active set) at a specified time.	NO	YES (Full / Snapshot)

effectiveTime	Time	The inclusive date or time at which this version of the identified reference set member became the current version. **Note** : In distribution files the effectiveTime should follow the short ISO date format (*YYYYMMDD*) and should not include the hours, minutes, seconds or timezone indicator. The current version of this reference set member at time *T* is the version with the most recent effectiveTime prior to or equal to time *T* .	YES	YES (Full) Optional (Snapshot)
active	Boolean	The state of the identified reference set member as at the specified effectiveTime . If active = 1 (true) the reference set member is part of the current version of the set, if active = 0 (false) the reference set member is not part of the current version of the set.	YES	NO
moduleId	SCTID	Identifies the SNOMED CT module that contains this reference set member as at the specified effectiveTime . The value must be a subtype of 900000000000443000 | Module (core metadata concept) | within the metadata hierarchy .	YES	NO
refsetId	SCTID	Identifies the reference set to which this reference set member belongs. In this case, a subtype descendant of: 900000000000480006 | Attribute value type reference set |	NO	NO
referencedComponentId	SCTID	A reference to the SNOMED CT component to be included in the reference set .	NO	NO
valueId	SCTID	The tagged value applied to the referencedComponentId . A subtype of 900000000000491004 | Attribute value | .	Depends on specific use	NO

Metadata

The metadata concepts shown in Table 5.2.3-2 are examples of concepts that identify attribute value reference sets. Table 5.2.3-2 and Table 5.2.3-3 show the metadata concepts that represent the permitted values for the valueId column in the |Concept inactivation indicator reference set| and |Description inactivation indicator reference set|.

Table 5.2.3-2: Attribute Value Reference Sets in the Metadata Hierarchy

```
900000000000454005 |Foundation metadata concept|
   900000000000455006 |Reference set|
       900000000000480006 |Attribute value type|
           900000000000489007 |Concept inactivation indicator reference set|
           900000000000490003 |Description inactivation indicator reference set|
           900000000000547002 |Relationship inactivation indicator reference set|  /*  <-- Not currently used  */
          /*  Other attribute value reference sets exist but are not used to track component inactivation  */
```

Item 5.2.3-1: Concept Inactivation Values (with usage notes)

```
900000000001043018 |Concept inactivation value|
   723277005 |Nonconformance to editorial policy component|  /*  <--
New value introduced in 2017-07-31 International Release  */
   900000000000482003 |Duplicate|
   900000000000483008 |Outdated|
   900000000000484002 |Ambiguous|
   900000000000485001 |Erroneous|
   900000000000486000 |Limited|
   900000000000487009 |Moved elsewhere|
   900000000000492006 |Pending move|    /* <-- NEVER used for descriptions in the International Release -
may have been used in extensions  */
```

Table 5.2.3-3: Description Inactivation Values (with usage notes)

```
  900000000001077011 |Description inactivation value|
     723277005 |Nonconformance to editorial policy component|  /* <--
New value introduced in 2017-07-31 International Release  */
     723278000 |Not semantically equivalent component|      /* <--
New value introduced in 2017-07-31 International Release  */
     900000000000483008 |Outdated|
     900000000000485001 |Erroneous|
     900000000000495008 |Concept non-current|
     900000000000486000 |Limited|       /* <--
NOT used for description inactivations after 2010-07-31 International Releases  */
     900000000000487009 |Moved elsewhere|  /* <--
NOT used for description inactivations before 2016-07-31 or after 2017-07-31 International Releases  */
     900000000000482003 |Duplicate|      /* <--
NOT used for description inactivations before 2016-07-31 or after 2017-07-31 International Releases  */
     900000000000494007 |Inappropriate|   /* <--
NOT used for description inactivations before 2008-07-31 or after 2017-07-31 International Releases  */
     900000000000492006 |Pending move|    /* <-- NEVER used for descriptions in the International Release -
may have been used in extensions  */
```

Reference Set Descriptor and Example Data

Notes on the tables used to show descriptors and examples
The reference set example tables on this page have been revised as follows to aid clarity and understanding:

- The first four columns which are present in all release files are not shown. The omitted columns (id, effectiveTime, active, moduleId) are used in the same way in all referenced sets to support identification, versioning and packaging. They do not directly affect the specific features of a particular reference set or reference set type.
- Reference set columns that contain SNOMED CT identifiers are expanded to show details of the concept or description referenced by that identifier. In some cases, the term is shown in the same column using the expression syntax, in other cases an additional column with a name suffix '_term' has been added. In the standard reference set files only the identifier is present in the column and there is no added column for the term. When using reference sets, the term and other details of the component are looked up from the relevant component release files.

Descriptor Template

The tables below show the descriptors that define examples of reference sets that follow the 900000000000480006 |Attribute value type reference set| pattern.

Table 5.2.3-5: Refset Descriptor Rows for the Concept Inactivation Indicator Reference Set

refsetId	referencedComponentId (Referenced component)	attributeDescription (Attribute description)	attributeType (Attribute type)	attributeOrder (Attribute order)
900000000000456007 | Reference set descriptor|	900000000000489007 | Concept inactivation indicator reference set|	449608002 |Referenced component|	900000000000461009 | Concept type component |	0
900000000000456007 | Reference set descriptor|	900000000000489007 | Concept inactivation indicator reference set|	900000000000481005 | Concept inactivation value|	900000000000461009 | Concept type component |	1

Table 5.2.3-5: Refset Descriptor Rows for the Description Inactivation Indicator Reference Set

refsetId	referencedComponentId (Referenced component)	attributeDescription (Attribute description)	attributeType (Attribute type)	attributeOrder (Attribute order)
900000000000456007 | Reference set descriptor|	900000000000490003 | Description inactivation indicator reference set|	449608002 |Referenced component|	900000000000462002 | Description type component|	0
900000000000456007 | Reference set descriptor|	900000000000490003 | Description inactivation indicator reference set|	900000000000493001 | Description inactivation value|	900000000000461009 | Concept type component |	1

Example Data

Example 5.2.3-1: Sample Rows from the Concept Inactivation Indicator Reference Set

refsetId	referencedComponentId (Referenced component)	valueId (Concept inactivation value)
900000000000489007 |Concept inactivation indicator reference set|	105000 |Poisoning by pharmaceutical excipient |	900000000000482003 |Duplicate|
900000000000489007 |Concept inactivation indicator reference set|	123008 |Channel catfish virus disease|	900000000000487009 |Moved elsewhere|
900000000000489007 |Concept inactivation indicator reference set|	141000 |Glaucoma as birth trauma|	900000000000482003 |Duplicate|
900000000000489007 |Concept inactivation indicator reference set|	157000 |AIDS with low vision|	900000000000484002 |Ambiguous|
900000000000489007 |Concept inactivation indicator reference set|	190000 |Partial hysterectomy|	900000000000484002 |Ambiguous|
900000000000489007 |Concept inactivation indicator reference set|	203004 |Replacement of pacemaker in brain|	900000000000484002 |Ambiguous|
900000000000489007 |Concept inactivation indicator reference set|	212002 |Salmonella III arizonae 53:k:z|	900000000000483008 |Outdated|
900000000000489007 |Concept inactivation indicator reference set|	215000 |Operative procedure on fingers|	900000000000482003 |Duplicate|

refsetId	referencedComponentId (Referenced component)	valueId (Concept inactivation value)
900000000000489007 |Concept inactivation indicator reference set|	220000 |Unspecified monoarthritis|	900000000000486000 |Limited|
900000000000489007 |Concept inactivation indicator reference set|	236003 |Incision of vein|	900000000000484002 |Ambiguous|

5.2.4 Language Reference Set

Purpose

A 900000000000506000 |Language type reference set| supports the representation of language and dialects preferences for the use of particular descriptions. The most common use case for this type of reference set is to specify the acceptable and preferred terms for use within a particular country or region. However, the same type of reference set can also be used to represent preferences for use of descriptions in a more specific context such as a clinical specialty, organization or department.

Data structure

A Language reference set is a Component reference set that is used to indicate which descriptions contain terms that are acceptable or preferred in a particular language or dialect . Its structure is shown in the following table.

Table 5.2.4-5: Language reference set - Data structure

Field	Data type	Purpose	Mutable	Part of Primary Key
id	UUID	A 128 bit unsigned Integer, uniquely identifying this reference set member. Different versions of a *reference set member* share the same id but have different effectiveTime. This allows a *reference set member* to be modified or made inactive (i.e. removed from the active set) at a specified time.	NO	YES (Full / Snapshot)
effectiveTime	Time	The inclusive date or time at which this version of the identified reference set member became the current version. **Note** : In distribution files the effectiveTime should follow the short ISO date format (*YYYYMMDD*) and should not include the hours, minutes, seconds or timezone indicator. The current version of this reference set member at time *T* is the version with the most recent effectiveTime prior to or equal to time *T* .	YES	YES (Full) Optional (Snapshot)
active	Boolean	The state of the identified reference set member as at the specified effectiveTime . If active = 1 (true) the reference set member is part of the current version of the set, if active = 0 (false) the reference set member is not part of the current version of the set.	YES	NO
moduleId	SCTID	Identifies the SNOMED CT module that contains this reference set member as at the specified effectiveTime . The value must be a subtype of 900000000000443000 | Module (core metadata concept) | within the metadata hierarchy.	YES	NO

refsetId	SCTID	Identifies the reference set to which this reference set member belongs. In this case, a subtype descendant of: 900000000000506000 | Language type |	NO	NO
referencedComponentId	SCTID	A reference to the SNOMED CT component to be included in the reference set. Refers to the description to which the acceptability value is being applied.	NO	NO
acceptabilityId	SCTID	A subtype of 900000000000511003 | Acceptability | indicating whether the description is acceptable or preferred for use in the specified language or dialect .	YES	NO

Language Reference Rules and Guidance

In a Language reference set:

- No more than one description of a specific description type associated with a single concept may have the acceptabilityId value 900000000000548007 |Preferred|.
- Every active concept should have one preferred synonym in each language.
  - This means that a language reference set should assign the acceptabilityId 900000000000548007 | Preferred| to one synonym (a description with typeId value 900000000000013009 |synonym|) associated with each concept .
  - This description is the preferred term for that concept in the specified language or dialect.
- Any description which is not referenced by an active row in the reference set is regarded as unacceptable (i.e. not a valid synonym in the language or dialect).
  - If a description becomes unacceptable, the relevant language reference set member is inactivated by adding a new row with the same id, the effectiveTime of the the change and the value active=0.
  - For this reason there is no requirement for an "unacceptable" value.

Metadata

The following metadata supports this reference set :

Table 5.2.4-5: Language References Sets in the Metadata Hierarchy

- 900000000000454005 |Foundation metadata concept|
  - 900000000000506000 |Language type|
    - 900000000000507009 |English|
      - 900000000000508004 |GB English|
      - 900000000000509007 |US English|

The immediate children of |Language type| will represent languages. This level may be used to represent the "formal approved" language, where a language authority is formally recognized. In most cases, this level will not identify a specific reference set. Subtype of the language level are used to represent different dialects, national or regional variants.

Reference Set Descriptor and Example Data

Notes on the tables used to show descriptors and examples
The reference set example tables on this page have been revised as follows to aid clarity and understanding:

- The first four columns which are present in all release files are not shown. The omitted columns (id, effectiveTime, active, moduleId) are used in the same way in all referenced sets to support identification, versioning and packaging. They do not directly affect the specific features of a particular reference set or reference set type.
- Reference set columns that contain SNOMED CT identifiers are expanded to show details of the concept or description referenced by that identifier. In some cases, the term is shown in the same column using the expression syntax, in other cases an additional column with a name suffix '_term' has been added. In the standard reference set files only the identifier is present in the column and there is no added column for the term. When using reference sets, the term and other details of the component are looked up from the relevant component release files.

Language Reference Descriptor

The table below shows an example of the descriptor for a specific reference sets that follows the 900000000000506000 |Language type reference set| pattern.

Table 5.2.4-5: Refset Descriptor rows for a language reference set

refsetId	referencedComponentId (Referenced component)	attributeDescription (Attribute description)	attributeType (Attribute type)	attributeOrder (Attribute order)
900000000000456007 |Reference set descriptor|	900000000000508004 |GB English|	900000000000510002 |Description in dialect|	900000000000462002 |Description type component|	0
900000000000456007 |Reference set descriptor|	900000000000508004 |GB English|	900000000000511003 |Acceptability|	900000000000461009 |Concept type component |	1

Language Reference Set Examples

Table 5.2.4-5: Sample content from the US English language reference set

refsetId	referencedComponentId (Description)	acceptabilityId (Acceptability in dialect)
900000000000509007 |US English|	42969009 |Cauterization of skin {id:71693012}|	900000000000548007 |Preferred|
900000000000509007 |US English|	42969009 |Fulguration of subcutaneous tissue {id: 71695017}|	900000000000549004 |Acceptable|
900000000000509007 |US English|	80146002 |Appendectomy {id:132967011}|	900000000000548007 |Preferred|
900000000000509007 |US English|	80146002 |Excision of appendix {id:132972019}|	900000000000549004 |Acceptable|
900000000000509007 |US English|	271737000 |Anemia {id:406636013}|	900000000000548007 |Preferred|
900000000000509007 |US English|	271737000 |Absolute anemia {id:406640016}|	900000000000549004 |Acceptable|

Table 5.2.4-5: Sample content from the GB English language reference set

refsetId	**referencedComponentId (Description)**	**acceptabilityId (Acceptability in dialect)**
900000000000508004 |GB English|	42969009 |Cauterisation of skin {id:493493018}|	900000000000548007 |Preferred|
900000000000508004 |GB English|	42969009 |Fulguration of subcutaneous tissue {id: 71695017}|	900000000000549004 |Acceptable|
900000000000508004 |GB English|	80146002 |Excision of appendix {id:132972019}|	900000000000549004 |Acceptable|
900000000000508004 |GB English|	80146002 |Appendicectomy {id:132973012}|	900000000000548007 |Preferred|
900000000000508004 |GB English|	271737000 |Anaemia {id:406638014}|	900000000000548007 |Preferred|
900000000000508004 |GB English|	271737000 |Absolute anaemia {id:406641017}|	900000000000549004 |Acceptable|

In the above examples, 80146002 |Excision of appendix| is acceptable in both US and GB English. However, 80146002 |Appendectomy| is preferred in US English and 80146002 |Appendicectomy| is preferred in GB English.

5.2.5 Association Reference Set

Purpose

An 900000000000521006 |Association type reference set| represents a set of unordered associations of a particular type between components.

Data structure

An Association reference set is a reference set used to represent associations between components. Its structure is shown in the following table.

Table 5.2.5-5: Association reference Set - Data structure

Field	**Data type**	**Purpose**	Mutable	**Part of Primary Key**
id	UUID	A 128 bit unsigned Integer, uniquely identifying this reference set member. Different versions of a *reference set member* share the same id but have different effectiveTime. This allows a *reference set member* to be modified or made inactive (i.e. removed from the active set) at a specified time.	NO	YES (Full / Snapshot)
effectiveTime	Time	The inclusive date or time at which this version of the identified reference set member became the current version. **Note** : In distribution files the effectiveTime should follow the short ISO date format (*YYYYMM D D*) and should not include the hours, minutes, seconds or timezone indicator. The current version of this reference set member at time *T* is the version with the most recent effectiveTime prior to or equal to time *T* .	YES	YES (Full) Optional (Snapshot)
active	Boolean	The state of the identified reference set member as at the specified effectiveTime . If active = 1 (true) the reference set member is part of the current version of the set, if active = 0 (false) the reference set member is not part of the current version of the set.	YES	NO

moduleId	SCTID	Identifies the SNOMED CT module that contains this reference set member as at the specified effectiveTime . The value must be a subtype of 900000000000443000 | Module (core metadata concept) | within the metadata hierarchy .	YES	NO
refsetId	SCTID	Identifies the reference set to which this reference set member belongs. In this case, a subtype descendant of: 900000000000521006 | Association type |	NO	NO
referencedComponentId	SCTID	A reference to the SNOMED CT component to be included in the reference set . The source component of the association.	NO	NO
targetComponentId	SCTID	The identifier of the target component of the association. An inconsistency in this specification was resolved by the Modeling Advisory Group 2018-10-15 decision that this field should be mutable.	YES	NO

Metadata

The following metadata supports this reference set:

Table 5.2.5-5: Association Reference Sets in the Metadata Hierarchy

```
900000000000455006 |Reference set|
   900000000000521006 |Association type|
      900000000000522004 |Historical association|
         900000000000523009 |POSSIBLY EQUIVALENT TO association reference set|
         900000000000524003 |MOVED TO association reference set|
         900000000000525002 |MOVED FROM association reference set|
         900000000000526001 |REPLACED BY association reference set|
         900000000000527005 |SAME AS association reference set|
         900000000000528000 |WAS A association reference set|
         900000000000529008 |SIMILAR TO association reference set|
         900000000000530003 |ALTERNATIVE association reference set|
         900000000000531004 |REFERS TO concept association reference set|
```

Historical Association Rules and Guidance

Each member of a 900000000000522004 |Historical association| reference set represents a reference from an inactive component to other equivalent or related components that were current in the Release Version in which that component was inactivated.

Each 900000000000522004 |Historical association| reference set represents a different type of association between the components referred to by the referencedComponentId and the targetComponentId as shown in Table 5.2.5-3.

Table 5.2.5-3: Association reference set types in the International Release of SNOMED CT

Association reference set	Descriptions
900000000000523009 |POSSIBLY EQUIVALENT TO association reference set|	Applies to a concept that is ambiguous. The targetComponent is an active concept that represents one of the possible meanings of the inactive concept . Multiple rows are used to refer to each of the possible meanings of the ambiguous concept.
900000000000524003 |MOVED TO association reference set|	Applies to a component that has been moved to (or are pending a move to) another namespace. The targetComponent identifies the target namespace (not the new component).
900000000000525002 |MOVED FROM association reference set|	Applies to a component that has been moved to this namespace from another namespace. The targetComponent identifies the original component Identifier in its previous namespace.
900000000000526001 |REPLACED BY association reference set|	Applies to an erroneous, obsolete and other inactive component for which there is a single active replacement. The targetComponent identifies the active component that replaces this component.
900000000000527005 |SAME AS association reference set|	Applies to a component that is a duplicate. The targetComponent identifies the active component that this component duplicates.
900000000000528000 |WAS A association reference set|	Links an inactive classification concept such as "not otherwise specified" or "otherwise specified" with the active concept that was formerly its most proximal supertype.
900000000000529008 |SIMILAR TO association reference set|	(not used currently)
900000000000530003 |ALTERNATIVE association reference set|	Links an inactive classification concept derived from ICD-9 Chapter XVI "Symptoms signs and ill-defined conditions" with the most similar active concept.
900000000000531004 |REFERS TO concept association reference set|	Applies to an inactive description which is inappropriate to the concept it is directly linked to but instead should refer to the concept referenced by the targetComponent.

The component identified by the targetComponentId must be an instance of the same class of component as the component identified by the referencedComponentId for all |Historical association| reference sets apart from the |REFERS TO concept association reference set|.

Reference Set Descriptor and Example Data

Notes on the tables used to show descriptors and examples
The reference set example tables on this page have been revised as follows to aid clarity and understanding:

- The first four columns which are present in all release files are not shown. The omitted columns (id, effectiveTime, active, moduleId) are used in the same way in all referenced sets to support identification, versioning and packaging. They do not directly affect the specific features of a particular reference set or reference set type.
- Reference set columns that contain SNOMED CT identifiers are expanded to show details of the concept or description referenced by that identifier. In some cases, the term is shown in the same column using the expression syntax, in other cases an additional column with a name suffix '_term' has been added. In the standard reference set files only the identifier is present in the column and there is no added column for the term. When using reference sets, the term and other details of the component are looked up from the relevant component release files.

Descriptor Template

The tables below show examples of the descriptors for specific reference sets that follow the 900000000000521006 | Association type reference set| pattern.

Table 5.2.5-5: Refset Descriptor rows for the SAME AS association reference set

refsetId	**referencedComponentId (Referenced component)**	**attributeDescription (Attribute description)**	**attributeType (Attribute type)**	**attributeOrder (Attribute order)**
900000000000456007 | Reference set descriptor|	900000000000527005 |SAME AS association reference set|	900000000000532006 | Association source component|	900000000000460005 | Component type|	0
900000000000456007 | Reference set descriptor|	900000000000527005 |SAME AS association reference set|	900000000000533001 | Association target component|	900000000000460005 | Component type|	1

Example Data

The following table holds example entries for the 900000000000526001 |REPLACED BY association reference set|.

Table 5.2.5-5: Sample content from

refsetId	**referencedComponentId (Association source component)**	**targetComponentId (Association target component)**
900000000000526001 |REPLACED BY association reference set|	100005 |SNOMED RT Concept|	138875005 |SNOMED CT Concept|
900000000000526001 |REPLACED BY association reference set|	212002 |Salmonella III arizonae 53:k:z|	398450001 |Salmonella IIIb 53:k:z|
900000000000526001 |REPLACED BY association reference set|	225005 |Special care of patient with contagious disease|	133895001 |Care of patient with infectious disease|
900000000000526001 |REPLACED BY association reference set|	244003 |Evans and Lloyd-Thomas syndrome|	66659007 |Normal variation in position|
900000000000526001 |REPLACED BY association reference set|	278009 |Epidural injection of neurolytic substance, lumbar|	17753007 |Epidural injection of neurolytic solution, lumbar|
900000000000526001 |REPLACED BY association reference set|	558000 |Other disorder of the neurohypophysis, NEC|	72442006 |Disorder of posterior pituitary|
900000000000526001 |REPLACED BY association reference set|	659001 |Peptostreptococcus anaerobius|	413524006 |Anaerococcus tetradius|
900000000000526001 |REPLACED BY association reference set|	696005 |Chronobiologic disorder|	387605007 |Abnormal chronobiologic state|
900000000000526001 |REPLACED BY association reference set|	700002 |Salmonella III arizonae 50:z4,z23,z32:--|	404619004 |Salmonella IIIa 50:z4,z23,z32:-|

refsetId	referencedComponentId (Association source component)	targetComponentId (Association target component)
900000000000526001 |REPLACED BY association reference set|	822000 |Salmonella arizonae 53:z4,z23:--|	13998005 |Salmonella IV 53:z4,z23:--|

Relevant References

- Practical Guide or Reference Sets 3.2.6.3.2. Representing Historical Associations
- Terminology Services Guide 4.2.3 Historical Association Reference Sets.

5.2.6 Ordered Association Reference Set

Purpose

An 733618005 |Ordered association type reference set (foundation metadata concept)| can be used to specify ordered associations between different components. These can be used to specify several interrelated subsets of components and to define alternative hierarchies for navigation while searching for an appropriate concept or description.

Data structure

An Ordered association reference set is a component integer reference set that is used to represent ordered lists of associations and alternative hierarchies. Its structure is shown in the following table.

Table 5.2.6-3: Ordered association reference set - Data structure

Field	Data type	Purpose	Mutable	Part of Primary Key
id	UUID	A 128 bit unsigned Integer, uniquely identifying this reference set member. Different versions of a *reference set member* share the same id but have different effectiveTime. This allows a *reference set member* to be modified or made inactive (i.e. removed from the active set) at a specified time.	NO	YES (Full / Snapshot)
effectiveTime	Time	The inclusive date or time at which this version of the identified reference set member became the current version. **Note** : In distribution files the effectiveTime should follow the short ISO date format (*YYYYMM D D*) and should not include the hours, minutes, seconds or timezone indicator. The current version of this reference set member at time *T* is the version with the most recent effectiveTime prior to or equal to time *T* .	YES	YES (Full) Optional (Snapshot)
active	Boolean	The state of the identified reference set member as at the specified effectiveTime . If active = 1 (true) the reference set member is part of the current version of the set, if active = 0 (false) the reference set member is not part of the current version of the set.	YES	NO
moduleId	SCTID	Identifies the SNOMED CT module that contains this reference set member as at the specified effectiveTime . The value must be a subtype of 900000000000443000 | Module (core metadata concept) | within the metadata hierarchy.	YES	NO

refsetId	SCTID	Identifies the reference set to which this reference set member belongs. In this case, a subtype descendant of: 447258008 | Ordered type reference set |	NO	NO
referencedComponentId	SCTID	A reference to the SNOMED CT component to be included in the reference set. Refers to the source component of the association.	NO	NO
targetComponentId	SCTID	The identifier of the target component of the association that acts as a grouper or hierarchy node, collecting together a subgroup from within the list. This field either enables reference set member linked into a number of subgroups. These subgroups can be nested allowing representation of alternative hierarchies. To link members into a subgroup, all components in the same subgroup should reference the same component. This can either be a component that represents the name of that subgroup or the first member of the subgroup. In the latter case, the first row of each subgroup will contain the same identifier in referencedComponentId and targetComponentId and with order =1. To link a number of children concepts to a single parent concept, one member record should exist per child, with the referencedComponentId field referencing the parent and this field referencing the child concept. The order field is then used to order the children concepts under the parent concept.	NO	NO
order	integer	Specifies the sort order of the list. The list is ordered by applying an ascending sort of the order value. The value of order =1 represents the highest priority. A value of '0' is not allowed. Duplicate values are permitted and the sort order between two members with the same order value is not defined. If the targetComponentId value is not 0, sorting occurs within subgroups that share the same targetComponentId. Note: The name "order" is a reserved word in some database environments. Please consider this when using this column.	YES	NO

Metadata

The following metadata in the "Foundation metadata concept" hierarchy supports this reference set:

Table 5.2.6-3: Ordered Association References Set in the Metadata Hierarchy

- 900000000000454005 |Foundation metadata concept|
  - 900000000000455006 |Reference set|
    - 733618005 |Ordered association type reference set (foundation metadata concept)|

Reference Set Descriptor and Example Data

Notes on the tables used to show descriptors and examples
The reference set example tables on this page have been revised as follows to aid clarity and understanding:

- The first four columns which are present in all release files are not shown. The omitted columns (id, effectiveTime, active, moduleId) are used in the same way in all referenced sets to support identification, versioning and packaging. They do not directly affect the specific features of a particular reference set or reference set type.
- Reference set columns that contain SNOMED CT identifiers are expanded to show details of the concept or description referenced by that identifier. In some cases, the term is shown in the same column using the expression syntax, in other cases an additional column with a name suffix '_term' has been added. In the standard reference set files only the identifier is present in the column and there is no added column for the term. When using reference sets, the term and other details of the component are looked up from the relevant component release files.

Descriptor Template

The tables below show the descriptor that defines the structure of the 447258008 |Ordered type reference set| pattern and an example of descriptor for a specific reference set that follows this pattern.

Table 5.2.6-3: Refset Descriptor rows for an ordered association type reference set

refsetId	referencedComponentId	attributeDescription	attributeType	attributeOrder
900000000000456007 |Reference set descriptor|	733618005 |Ordered association type reference set (foundation metadata concept)|	449608002 |Referenced component|	900000000000460005 |Component type|	0
900000000000456007 |Reference set descriptor|	733618005 |Ordered association type reference set (foundation metadata concept)|	900000000000533001 |Association target component|	900000000000460005 |Component type|	1
900000000000456007 |Reference set descriptor|	733618005 |Ordered association type reference set (foundation metadata concept)|	447255006 |Priority order reference set attribute|	900000000000478000 |Unsigned integer|	2

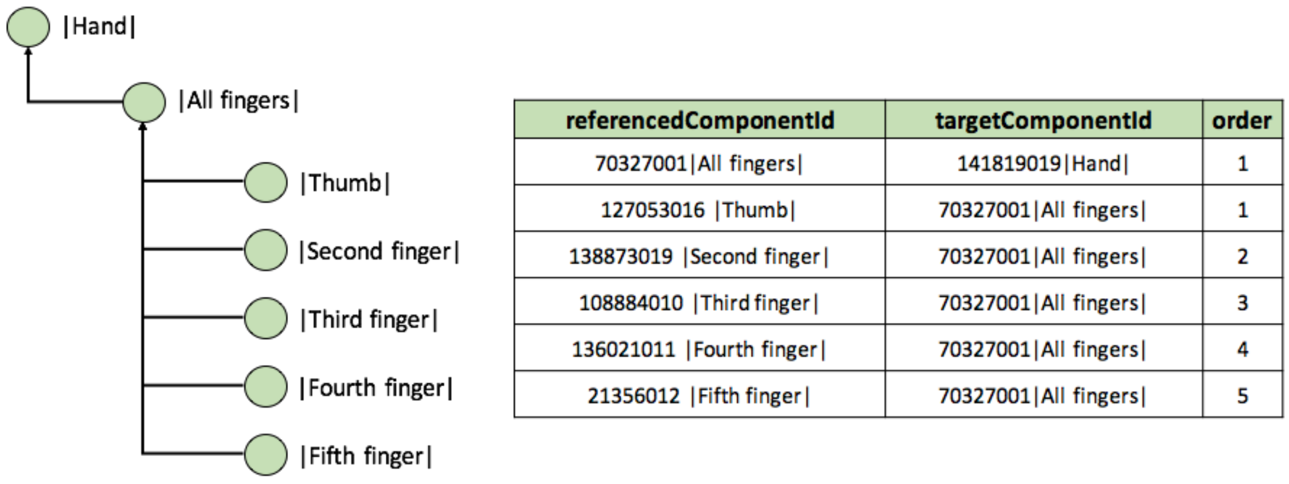
Example

Ordered association reference sets can be used to specify and display a customized navigation hierarchy. Alternative hierarchical representations of SNOMED CT can support data entry by satisfying the requirements of a specific use case, and addressing some of the challenges of displaying an unordered polyhierarchy (as defined by SNOMED CT's subtype structure).

The figure below shows the way a navigation hierarchy is represented. The example reference set contains a set of description components used to describe finger structures.

The | All fingers | components is linked to the | Hand |, and the | Thumb | is linked to the | All fingers component | The | Thumb | is placed first because it has the order value 1. Similarly, the components for | Second finger |, | Third finger |, | Fourth finger | and | Fifth finger | are also linked to the | All finger | component in the order specified by the order value. As shown in the figure the direction of the associations goes from the referenceComponentId to the linkedToId, so the components referenced by the linkedToId are used to form the groups specified in the hierarchy

id	effectiveTime	active	moduleId	refsetId	refsetId_term	referencedComponentId	referencedComponentId_term	targetComponentId	targetComponentId_term	order
…	20160731	1	19999999103	159999999105	Associations as ordered reference set	70327001	All fingers	141819019	Hand	1
…	20160731	1	19999999103	159999999105	Associations as ordered reference set	127053016	Thumb	70327001	All fingers	1
…	20160731	1	19999999103	159999999105	Associations as ordered reference set	138873019	Second finger	70327001	All fingers	2
…	20160731	1	19999999103	159999999105	Associations as ordered reference set	108884010	Third finger	70327001	All fingers	3
…	20160731	1	19999999103	159999999105	Associations as ordered reference set	136021011	Fourth finger	70327001	All fingers	4
…	20160731	1	19999999103	159999999105	Associations as ordered reference set	21356012	Fifth finger	70327001	All fingers	5



referencedComponentId	targetComponentId	order
70327001 |All fingers|	141819019 |Hand|	1
127053016 |Thumb|	70327001 |All fingers|	1
138873019 |Second finger|	70327001 |All fingers|	2
108884010 |Third finger|	70327001 |All fingers|	3
136021011 |Fourth finger|	70327001 |All fingers|	4
21356012 |Fifth finger|	70327001 |All fingers|	5

Figure 3.2.1.5-1: Navigation hierarchy example.

5.2.7 Annotation Reference Set

Purpose

An 900000000000516008 |Annotation type reference set| allows String to be associated with components for any specified purpose.

Data structure

An annotation reference set String reference set used to apply text annotation to selected SNOMED CT components .

Table 5.2.7-5: Annotation reference set - Data structure

Field	Data type	Purpose	Mutable	Part of Primary Key
id	UUID	A 128 bit unsigned Integer, uniquely identifying this reference set member. Different versions of a *reference set member* share the same id but have different effectiveTime. This allows a *reference set member* to be modified or made inactive (i.e. removed from the active set) at a specified time.	NO	YES (Full / Snapshot)
effectiveTime	Time	The inclusive date or time at which this version of the identified reference set member became the current version. **Note** : In distribution files the effectiveTime should follow the short ISO date format (*YYYYMM D D*) and should not include the hours, minutes, seconds or timezone indicator. The current version of this reference set member at time *T* is the version with the most recent effectiveTime prior to or equal to time *T* .	YES	YES (Full) Optional (Snapshot)
active	Boolean	The state of the identified reference set member as at the specified effectiveTime . If active = 1 (true) the reference set member is part of the current version of the set, if active = 0 (false) the reference set member is not part of the current version of the set.	YES	NO
moduleId	SCTID	Identifies the SNOMED CT module that contains this reference set member as at the specified effectiveTime . The value must be a subtype of 900000000000443000 | Module (core metadata concept) | within the metadata hierarchy.	YES	NO
refsetId	SCTID	Identifies the reference set to which this reference set member belongs. In this case, a subtype descendant of: 900000000000516008 | Annotation type |	NO	NO
referencedComponentId	SCTID	A reference to the SNOMED CT component to be included in the reference set. The component to which the annotation is being applied.	NO	NO
annotation	String	The text annotation to attach to the component identified by referencedComponentId .	YES	NO

Metadata

The following metadata in supports this reference set :

Table 5.2.7-5: Annotation References Sets in the Metadata Hierarchy

```
900000000000454005 |Foundation metadata concept|
    900000000000455006 |Reference set|
        900000000000516008 |Annotation type|
            900000000000517004 |Associated image|
```

Reference Set Descriptor and Example Data

Notes on the tables used to show descriptors and examples
The reference set example tables on this page have been revised as follows to aid clarity and understanding:

- The first four columns which are present in all release files are not shown. The omitted columns (id, effectiveTime, active, moduleId) are used in the same way in all referenced sets to support identification, versioning and packaging. They do not directly affect the specific features of a particular reference set or reference set type.
- Reference set columns that contain SNOMED CT identifiers are expanded to show details of the concept or description referenced by that identifier. In some cases, the term is shown in the same column using the expression syntax, in other cases an additional column with a name suffix '_term' has been added. In the standard reference set files only the identifier is present in the column and there is no added column for the term. When using reference sets, the term and other details of the component are looked up from the relevant component release files.

Descriptor Template

The tables below show the descriptors that define the structure of the 900000000000516008 |Annotation type reference set| pattern and examples of the descriptors for specific reference sets that follow this pattern.

Table 5.2.7-5: Descriptor Template for Annotation Reference Sets

refsetId	referencedComponentId	attributeDescription	attributeType	attributeOrder
900000000000456007 | Reference set descriptor|	900000000000516008 | Annotation type|	900000000000518009 | Annotated component|	900000000000461009 | Concept type component|	0
900000000000456007 | Reference set descriptor|	900000000000516008 | Annotation type|	900000000000519001 | Annotation|	900000000000465000 | String|	1

The attributeType for the Annotation field can be any descendant of the " string " concept in the metadata hierarchy. This hierarchy is described in more detail under the " Reference set descriptor" section.

Table 5.2.7-5: Descriptor for the Associated Image Annotation Reference set

refsetId	referencedComponentId	attributeDescription	attributeType	attributeOrder
900000000000456007 | Reference set descriptor|	900000000000517004 | Associated image|	900000000000518009 | Annotated component|	900000000000461009 | Concept type component|	0
900000000000456007 | Reference set descriptor|	900000000000517004 | Associated image|	900000000000520007 | Image|	900000000000469006 |URL|	1

Note that in the table above, the 900000000000469006 |URL|concept is a descendant of | string | concept in the metadata.

Annotation Reference Set Example

As no annotation reference sets are included in the International Release, these sample rows are for illustration only.

Table 5.2.7-5: Example of Associated image Annotation Reference Set

refsetId	referencedComponentId	Annotation
900000000000517004 |Associated image|	80891009 |Heart structure|	http://en.wikipedia.org/wiki/Heart#mediaviewer/File:Wiki_Heart_Antomy_Ties_van_Brussel.jpg
900000000000517004 |Associated image|	86174004 |Laparoscope|	http://www.educationaldimensions.com/eLearn/endoscope/bigScope.html

In the above example, the two URLs have been used to annotate two SNOMED CT concepts with images on the web. It is not recommended that this mechanism be used to annotate concepts with text that may require translation to other languages. Instead, such text should be included under an appropriate description type within the Description.

5.2.8 Query Specification Reference Set

Purpose

A 900000000000512005 |Query specification type reference set| allows a serialised query to represent the membership of a subset of SNOMED CT components. A query contained in the reference set is run against the content of SNOMED CT to produce a subset of concepts, descriptions or relationships. The query is referred to an intensional definition of the subset. It can be run against future releases of SNOMED CT to generate an updated set of subset members.

The members of the resulting subset may also be represented in an enumerated form as a Simple reference set . An enumerated representation of a subset is referred to as an extensional definition.

Data structure

A Query specification reference set is a String reference set containing query that represent intensional definitions of subsets of components. The result of running the query is an extensional representation of the subset of components which can be represented as a Simple reference set . Its structure is shown in the following table.

Table 5.2.8-4: Query specification reference set - Data structure

Field	Data type	Purpose	Mutable	Part of Primary Key
id	UUID	A 128 bit unsigned Integer , uniquely identifying this reference set member . Different versions of a *reference set member* share the same id but have different effectiveTime . This allows a *reference set member* to be modified or made inactive (i.e. removed from the active set) at a specified time.	NO	YES (Full / Snapshot)
effectiveTime	Time	The inclusive date or time at which this version of the identified reference set member became the current version. **Note** : In distribution files the effectiveTime should follow the short ISO date format (*YYYYMM DD* and should not include the hours, minutes, seconds or timezone indicator. The current version of this reference set member at time *T* is the version with the most recent effectiveTime prior to or equal to time *T* .	YES	YES (Full) Optional (Snapshot)
active	Boolean	The state of the identified reference set member as at the specified effectiveTime . If active = 1 (true) the reference set member is part of the current version of the set, if active = 0 (false) the reference set member is not part of the current version of the set.	YES	NO

moduleId	SCTID	Identifies the SNOMED CT module that contains this reference set member as at the specified effectiveTime . The value must be a subtype of 900000000000443000 | Module (core metadata concept) | within the metadata hierarchy.	YES	NO
refsetId	SCTID	Identifies the reference set to which this reference set member belongs. In this case, a subtype descendant of: 900000000000512005 | Query specification type |	NO	NO
referencedComponentId	SCTID	A reference to the SNOMED CT component to be included in the reference set.	NO	NO
query	String	The serialised query that can be used to (re-)generate the reference set members. • The syntax for specifying the intensional definitions of concept subsets is specified in the Error rendering macro 'sp-nobody-link' Conversion context did not contain original content entity. .	YES	NO

Metadata

The following metadata in the "Foundation metadata concept " hierarchy supports this reference set :

Table 5.2.8-4: Hierarchy of Foundation metadata concept

```
900000000000454005 |Foundation metadata concept|
   900000000000455006 |Reference set|
       900000000000512005 |Query specification type|
           900000000000513000 |Simple query specification|
```

Reference Set Descriptor and Example Data

Notes on the tables used to show descriptors and examples
The reference set example tables on this page have been revised as follows to aid clarity and understanding:

- The first four columns which are present in all release files are not shown. The omitted columns (id, effectiveTime, active, moduleId) are used in the same way in all referenced sets to support identification, versioning and packaging. They do not directly affect the specific features of a particular reference set or reference set type.
- Reference set columns that contain SNOMED CT identifiers are expanded to show details of the concept or description referenced by that identifier. In some cases, the term is shown in the same column using the expression syntax, in other cases an additional column with a name suffix '_term' has been added. In the standard reference set files only the identifier is present in the column and there is no added column for the term. When using reference sets, the term and other details of the component are looked up from the relevant component release files.

Descriptor Template

The table below shows the descriptor that defines the structure of the 900000000000512005 |Query specification type reference set| pattern.

Table 5.2.8-4: Descriptor Template for Query Specification Reference Sets

refsetId	referencedComponentId	attributeDescription	attributeType	attributeOrder
900000000000456007 |Reference set descriptor|	900000000000512005 |Query specification type reference set|	900000000000514006 |Generated reference set|	900000000000461009 |Concept type component|	0
900000000000456007 |Reference set descriptor|	900000000000512005 |Query specification type reference set|	900000000000515007 |Query|	900000000000465000 |String|	1

Example Data

In the example below, "serialised query 1" is a text string that can be used to generate members for Reference set *1*, which is a simple member reference set (without any additional fields within its member records).

Table 5.2.8-4: Example rows from Query Specification Reference Set

refsetId	referencedComponentId	query
900000000000513000 |Simple query specification|	|Target reference set|	< 19829001 |disorder of lung| : 116676008 |associated morphology| = << 79654002 |edema|

References

- See Expression Constraint Language - Specification and Guide for details of the language used to specify intensional definitions of concept subsets.

5.2.9 Simple Map Reference Set

Purpose

A 900000000000496009 |Simple map reference set| allows representation of simple maps between SNOMED CT concepts and values in other code systems. No constraints are put on the number of coding schemes supported, the number of codes within a particular scheme mapped to by a single SNOMED CT concept or the number of SNOMED CT concepts mapping to a particular code. However, this type of reference set is usually only appropriate where there is a close "one-to-one" mapping between SNOMED CT concepts and coded values in another code system.

Data structure

A Simple map reference set is a String reference set used to represent one-to-one maps between SNOMED CT concepts and codes in another terminology, classification or code system. Its structure is shown in the following table.

Table 5.2.9-4: Simple map reference set - Data structure

Field	Data type	Purpose	Mutable	Part of Primary Key
id	UUID	A 128 bit unsigned Integer, uniquely identifying this reference set member. Different versions of a *reference set member* share the same id but have different effectiveTime. This allows a *reference set member* to be modified or made inactive (i.e. removed from the active set) at a specified time.	NO	YES (Full / Snapshot)
effectiveTime	Time	The inclusive date or time at which this version of the identified reference set member became the current version. **Note** : In distribution files the effectiveTime should follow the short ISO date format (*YYYYMM D* *D*) and should not include the hours, minutes, seconds or timezone indicator. The current version of this reference set member at time *T* is the version with the most recent effectiveTime prior to or equal to time *T* .	YES	YES (Full) Optional (Snapshot)
active	Boolean	The state of the identified reference set member as at the specified effectiveTime . If active = 1 (true) the reference set member is part of the current version of the set, if active = 0 (false) the reference set member is not part of the current version of the set.	YES	NO
moduleId	SCTID	Identifies the SNOMED CT module that contains this reference set member as at the specified effectiveTime . The value must be a subtype of 900000000000443000 | Module (core metadata concept) | within the metadata hierarchy.	YES	NO
refsetId	SCTID	Identifies the reference set to which this reference set member belongs. In this case, a subtype descendant of: 900000000000496009 | Simple map type reference set |	NO	NO
referencedComponentId	SCTID	A reference to the SNOMED CT component to be included in the reference set. Refers to the SNOMED CT concept that is mapped to and/or from the other terminology or code system.	NO	NO
mapTarget	String	The equivalent code in the other terminology, classification or code system.	YES	NO

Metadata

The following metadata hierarchy supports this reference set:

Table 5.2.9-4: Simple Map Reference Sets in the Metadata Hierarchy

```
900000000000454005 |Foundation metadata concept|
    900000000000455006 |Reference set|
        900000000000456007 |Reference set descriptor|
            900000000000496009 |Simple map|
            900000000000497000 |CTV3 simple map|
                900000000000498005 |SNOMED RT ID simple map|
```

Reference Set Descriptor and Example Data

Notes on the tables used to show descriptors and examples
The reference set example tables on this page have been revised as follows to aid clarity and understanding:

- The first four columns which are present in all release files are not shown. The omitted columns (id, effectiveTime, active, moduleId) are used in the same way in all referenced sets to support identification, versioning and packaging. They do not directly affect the specific features of a particular reference set or reference set type.
- Reference set columns that contain SNOMED CT identifiers are expanded to show details of the concept or description referenced by that identifier. In some cases, the term is shown in the same column using the expression syntax, in other cases an additional column with a name suffix '_term' has been added. In the standard reference set files only the identifier is present in the column and there is no added column for the term. When using reference sets, the term and other details of the component are looked up from the relevant component release files.

Descriptor template and examples

The tables below show the descriptors that define examples of reference sets that follow the 900000000000496009 | Simple map reference set| pattern.

Table 5.2.9-4: Refset Descriptor rows for

refsetId	referencedComponentId (Referenced component)	attributeDescription (Attribute description)	attributeType (Attribute type)	attributeOrder (Attribute order)
900000000000456007 |Reference set descriptor|	446608001 |ICD-O simple map reference set|	900000000000500006 |Map source concept|	900000000000461009 |Concept type component|	0
900000000000456007 |Reference set descriptor|	446608001 |ICD-O simple map reference set|	900000000000499002 |Scheme value|	900000000000465000 |String|	1

Simple Map Refset Examples

Table 5.2.9-4: Sample Content from a Simple Map Reference Set

refsetId	referencedComponentId (Map source concept)	mapTarget (Scheme value)
900000000000498005 |SNOMED RT ID simple map|	100005 |SNOMED RT Concept|	G-3000
900000000000498005 |SNOMED RT ID simple map|	101009 |Quilonia ethiopica|	L-55535
900000000000498005 |SNOMED RT ID simple map|	102002 |Hemoglobin Okaloosa|	F-D5972
900000000000498005 |SNOMED RT ID simple map|	103007 |Squirrel fibroma virus|	L-37904

refsetId	referencedComponentId (Map source concept)	mapTarget (Scheme value)
900000000000498005 |SNOMED RT ID simple map|	104001 |Excision of lesion of patella|	P1-18376
900000000000498005 |SNOMED RT ID simple map|	105000 |Poisoning by pharmaceutical excipient|	DD-82950
900000000000498005 |SNOMED RT ID simple map|	106004 |Structure of posterior carpal region|	T-D8602
900000000000498005 |SNOMED RT ID simple map|	107008 |Structure of fetal part of placenta|	T-F1102
900000000000498005 |SNOMED RT ID simple map|	108003 |Entire condylar emissary vein|	T-49723
900000000000498005 |SNOMED RT ID simple map|	109006 |Anxiety disorder of childhood OR adolescence|	D9-12000

5.2.10 Complex and Extended Map Reference Sets

Purpose

A 447250001 |Complex map type reference set| enables representation of maps where each SNOMED CT concept may map to one or more codes in a target scheme. The type of reference set supports the general set of mapping data required to enable a target code to be selected at run-time from a number of alternate codes. It supports target code selection by accommodating the inclusion of machine readable rules and/or human readable advice. An 609331003 |Extended map type reference set| adds an additional field to allow categorization of maps.

Data structure

A *Complex map reference set* is an Integer - Integer - String - String - String - Component reference set. The pattern is currently used for the map to ICD-9-CM. Its structure is as shown in the following table, with one exception - the table includes an additional field (mapCategoryId) which is not used for this type of map.

An *Extended map reference set* follows the same pattern but adds one additional column. It is an Integer - Integer - String - String - String - Component- Component reference set and this pattern is currently used for maps to ICD-10. Its structure is shown in the following table.

Table 5.2.10-5: Complex and Extended map reference sets - Data structures

Field	Data type	Purpose	Mutable	Part of Primary Key
id	UUID	A 128 bit unsigned Integer, uniquely identifying this reference set member. Different versions of a *reference set member* share the same id but have different effectiveTime. This allows a *reference set member* to be modified or made inactive (i.e. removed from the active set) at a specified time.	NO	YES (Full / Snapshot)

effectiveTime	Time	The inclusive date or time at which this version of the identified reference set member became the current version. **Note** : In distribution files the effectiveTime should follow the short ISO date format (*YYYYMMDD*) and should not include the hours, minutes, seconds or timezone indicator. The current version of this reference set member at time *T* is the version with the most recent effectiveTime prior to or equal to time *T* .	YES	YES (Full) Optional (Snapshot)
active	Boolean	The state of the identified reference set member as at the specified effectiveTime . If active = 1 (true) the reference set member is part of the current version of the set, if active = 0 (false) the reference set member is not part of the current version of the set.	YES	NO
moduleId	SCTID	Identifies the SNOMED CT module that contains this reference set member as at the specified effectiveTime . The value must be a subtype of 900000000000443000 | Module (core metadata concept) | within the metadata hierarchy.	YES	NO
refsetId	SCTID	Identifies the reference set to which this reference set member belongs. In this case, a subtype descendant of: 447250001 | Complex map type reference set | or 609331003 | Extended map type reference set |	NO	NO
referencedComponentId	SCTID	A reference to the SNOMED CT component to be included in the reference set. Refers to the SNOMED CT concept that is mapped to the other terminology or code system.	NO	NO
mapGroup	Integer	An Integer, grouping a set of complex map records from which one may be selected as a target code. Where a SNOMED CT concept maps onto 'n' target codes, there will be 'n' groups, each containing one or more complex map records.	YES	NO
mapPriority	Integer	Within a mapGroup, the mapPriority specifies the order in which complex map records should be checked. Only the first map record meeting the run - time selection criteria will be taken as the target code within the group of alternate codes.	YES	NO
mapRule	String	A machine-readable rule, (evaluating to either 'true' or 'false' at run-time) that indicates whether this map record should be selected within its mapGroup.	YES	NO
mapAdvice	String	Human-readable advice, that may be employed by the software vendor to give an end-user advice on selection of the appropriate target code from the alternatives presented to him within the group.	YES	NO
mapTarget	String	The target code in the target terminology, classification or code system.	YES	NO
correlationId	SCTID	A child of 447247004 | SNOMED CT source code to target map code correlation value | in the metadata hierarchy, identifying the correlation between the SNOMED CT concept and the target code.	YES	NO
The following additional field only applies to 609331003 | Extended map type reference set |				

mapCategoryId	SCTID	Identifies the SNOMED CT concept in the metadata hierarchy which represents the MapCategory for the associated map member. The categories vary for different target code systems, each set of categories is represented by a subtype of 609330002 | Map category value | . In the case of ICD-10 the individual category values are subtypes of: 447634004 | ICD-10 map category value | .	YES	NO

Map Group, Priority and Rules

Values for mapGroup are allocated on a sequential basis (for each refsetId and referencedComponentId combination) during authoring starting at 1. However, distributed mapGroup are not necessarily sequential, as some mapGroup may be created and removed during a mapping process between releases. For maps where each SNOMED CT concept only maps to at most one of a group of alternate target codes, the mapGroup field are usually be set to '1'.

Values for mapPriority will be allocated on a sequential basis (within each map group) starting from '1'. For maps that do not require run - time alternatives, the mapPriority field is set to '1'.

The mapRule and mapAdvice fields enable run-time selection (within vendor's software) from a number of alternative map records within a mapGroup . Where there are no alternatives maps these columns of the release files will be empty (zero length string). Where alternative maps exist one or both of columns will be populated where relevant information is available.

Where both fields are populated, and a vendor's system is capable of processing a machine readable rule, this should take priority over the human readable advice. Where neither field is populated, a vendor's system should allow the end-user to select the appropriate target code from the alternates.

For more details on this topic in relation to the ICD-10 maps released as part of the SNOMED CT International Edition please see the ICD-10 Mapping Technical Guide

Mapping Rule Specifications

The specific grammar and content of the rules for resolving complex mapping cases depends on the nature of the ta rget code system or classification. In general, each map is accompanied by a rule which is tested against other data and can be evaluated to return one of the following values:

- **True** - in which case the map target applies;
- **False** - in which case the map target does not apply;
- **Indeterminate** - in cases where there is insufficient accessible data to determine whether the map target applies. In this case manual resolution of the map using the map advice provided will be required.

The mapping rules assume access to a number of variables, that can be bound to appropriate attributes in the vendor's system information model. These include the age and gender of the patient and information about coexisting situations (e.g. records of other disorders, procedures or events in the same patient record).

Detailed definitions of the mapping rules used forms part of individual specifications for maps to particular target code systems and classifications. This will initially be provided separately and will accompany the release of the relevant mapping files. For example, the set of rules used for mapping to ICD-10 are included in the ICD-10 Mapping Technical Guide.

Metadata

The following metadata supports this reference set:

Table 5.2.10-5: Complex Map References Sets in the Metadata Hierarchy

```
900000000000454005 |Foundation metadata concept|
    900000000000455006 |Reference set|
```

447250001 |Complex map type reference set|
609331003 |Extended map type reference set|

Reference Set Descriptor and Example Data

Notes on the tables used to show descriptors and examples
The reference set example tables on this page have been revised as follows to aid clarity and understanding:

- The first four columns which are present in all release files are not shown. The omitted columns (id, effectiveTime, active, moduleId) are used in the same way in all referenced sets to support identification, versioning and packaging. They do not directly affect the specific features of a particular reference set or reference set type.
- Reference set columns that contain SNOMED CT identifiers are expanded to show details of the concept or description referenced by that identifier. In some cases, the term is shown in the same column using the expression syntax, in other cases an additional column with a name suffix '_term' has been added. In the standard reference set files only the identifier is present in the column and there is no added column for the term. When using reference sets, the term and other details of the component are looked up from the relevant component release files.

Descriptor Templates

The tables below examples of the descriptors for specific reference sets that follow the 447250001 |Complex map type reference set|and 609331003 |Extended map type reference set|patterns.

Table 5.2.10-5: Refset Descriptor Rows for a Complex Map Reference Set

refsetId	referencedComponentId (Referenced component)	attributeDescription (Attribute description)	attributeType (Attribute type)	attributeOrder (Attribute order)
900000000000456007 |Reference set descriptor|	447563008 |ICD-9-CM equivalence complex map reference set|	900000000000500006 |Map source concept|	900000000000461009 |Concept type component|	0
900000000000456007 |Reference set descriptor|	447563008 |ICD-9-CM equivalence complex map reference set|	900000000000501005 |Map group|	900000000000478000 |Unsigned integer|	1
900000000000456007 |Reference set descriptor|	447563008 |ICD-9-CM equivalence complex map reference set|	900000000000502003 |Map priority|	900000000000478000 |Unsigned integer|	2
900000000000456007 |Reference set descriptor|	447563008 |ICD-9-CM equivalence complex map reference set|	900000000000503008 |Map rule |	900000000000465000 |String|	3
900000000000456007 |Reference set descriptor|	447563008 |ICD-9-CM equivalence complex map reference set|	900000000000504002 |Map advice|	900000000000465000 |String|	4
900000000000456007 |Reference set descriptor|	447563008 |ICD-9-CM equivalence complex map reference set|	900000000000505001 |Map target|	900000000000465000 |String|	5

refsetId	referencedComponentId (Referenced component)	attributeDescription (Attribute description)	attributeType (Attribute type)	attributeOrder (Attribute order)
900000000000456007 |Reference set descriptor|	447563008 |ICD-9-CM equivalence complex map reference set|	447247004 |SNOMED CT source code to target map code correlation value|	900000000000461009 |Concept type component|	6

Table 5.2.10-5: Refset Descriptor Rows for an Extended Map Reference Set

refsetId	referencedComponentId (Referenced component)	attributeDescription (Attribute description)	attributeType (Attribute type)	attributeOrder (Attribute order)
900000000000456007 |Reference set descriptor|	447562003 |ICD-10 complex map reference set|	900000000000500006 |Map source concept|	900000000000461009 |Concept type component|	0
900000000000456007 |Reference set descriptor|	447562003 |ICD-10 complex map reference set|	900000000000501005 |Map group|	900000000000478000 |Unsigned integer|	1
900000000000456007 |Reference set descriptor|	447562003 |ICD-10 complex map reference set|	900000000000502003 |Map priority|	900000000000478000 |Unsigned integer|	2
900000000000456007 |Reference set descriptor|	447562003 |ICD-10 complex map reference set|	900000000000503008 |Map rule |	900000000000465000 |String|	3
900000000000456007 |Reference set descriptor|	447562003 |ICD-10 complex map reference set|	900000000000504002 |Map advice|	900000000000465000 |String|	4
900000000000456007 |Reference set descriptor|	447562003 |ICD-10 complex map reference set|	900000000000505001 |Map target|	900000000000465000 |String|	5
900000000000456007 |Reference set descriptor|	447562003 |ICD-10 complex map reference set|	447247004 |SNOMED CT source code to target map code correlation value|	900000000000461009 |Concept type component|	6
900000000000456007 |Reference set descriptor|	447562003 |ICD-10 complex map reference set|	609330002 |Map category value |	900000000000461009 |Concept type component|	7

Example Data

Table 5.2.10-5: Sample Content from an Extended Map Reference Set

refSetId	referencedComponentId (Map source concept)	mapGroup (Map group)	mapPriority (Map priority)	mapRule (Map rule)	mapAdvice (Map advice)	mapTarget (Map target)	correlationId (SNOMED CT source code to target map code correlation value)	mapCategoryId (Map category value)
447562003 | ICD-10 complex map reference set|	127009 | Miscarriage with laceration of cervix|	1	1	TRUE	ALWAYS O03.8	O03.8	447561005 |SNOMED CT source code to target map code correlation not specified|	447637006 | Map source concept is properly classified|
447562003 | ICD-10 complex map reference set|	127009 | Miscarriage with laceration of cervix|	2	1	TRUE	ALWAYS O08.6	O08.6	447561005 |SNOMED CT source code to target map code correlation not specified|	447637006 | Map source concept is properly classified|
447562003 | ICD-10 complex map reference set|	140004 | Chronic pharyngitis|	1	1	IFA 90979004 | Chronic tonsillitis (disorder)|	IF CHRONIC TONSILLITIS CHOOSE J35.0 MAP OF SOURCE CONCEPT IS CONTEXT DEPENDENT	J35.0	447561005 |SNOMED CT source code to target map code correlation not specified|	447639009 | Map of source concept is context dependent|
447562003 | ICD-10 complex map reference set|	140004 | Chronic pharyngitis|	1	2	IFA 232406009 | Chronic pharyngeal candidiasis (disorder)|	IF CHRONIC PHARYNGEAL CANDIDIASIS CHOOSE B37.8 MAP OF SOURCE CONCEPT IS CONTEXT DEPENDENT	B37.8	447561005 |SNOMED CT source code to target map code correlation not specified|	447639009 | Map of source concept is context dependent|
447562003 | ICD-10 complex map reference set|	140004 | Chronic pharyngitis|	1	3	OTHERWISE TRUE	ALWAYS J31.2	J31.2	447561005 |SNOMED CT source code to target map code correlation not specified|	447637006 | Map source concept is properly classified|
447562003 | ICD-10 complex map reference set|	162004 |Severe manic bipolar I disorder without psychotic features|	1	1	TRUE	ALWAYS F31.1	F31.1	447561005 |SNOMED CT source code to target map code correlation not specified|	447637006 | Map source concept is properly classified|
447562003 | ICD-10 complex map reference set|	177007 | Poisoning by sawfly larvae|	1	1	TRUE	ALWAYS T63.4	T63.4	447561005 |SNOMED CT source code to target map code correlation not specified|	447637006 | Map source concept is properly classified|
447562003 | ICD-10 complex map reference set|	177007 | Poisoning by sawfly larvae|	2	1	TRUE	ALWAYS X25	X25	447561005 |SNOMED CT source code to target map code correlation not specified|	447637006 | Map source concept is properly classified|

refSetId	referencedComponentId (Map source concept)	mapGroup (Map group)	mapPriority (Map priority)	mapRule (Map rule)	mapAdvice (Map advice)	mapTarget (Map target)	correlationId (SNOMED CT source code to target map code correlation value)	mapCategoryId (Map category value)
447562003 | ICD-10 complex map reference set|	181007 | Hemorrhagic bronchopneumonia|	1	1	TRUE	ALWAYS J18.0	J18.0	447561005 |SNOMED CT source code to target map code correlation not specified|	447637006 | Map source concept is properly classified|
447562003 | ICD-10 complex map reference set|	183005 | Autoimmune pancytopenia|	1	1	TRUE	ALWAYS D61.8	D61.8	447561005 |SNOMED CT source code to target map code correlation not specified|	447637006 | Map source concept is properly classified|

Related Links

- ICD-10 Mapping Technical Guide

5.2.11 Reference Set Descriptor

Purpose

The 900000000000456007 |Reference set descriptor| is a reference set that is used to specify the format of all reference sets included in a release. The data type and meaning of the referenced component and each additional field within each reference set is described by this reference set .

Reference set descriptor can be used to define

- The order of appearance of additional attributes (other than those mandatory for all reference sets);
- The name and purpose of the additional attributes;
- The data types for the additional attributes.

This allows for a reference set to be validated using the metadata embedded within the reference set descriptor in the following ways:

- the data type of its attributes may be validated against the data type declared in the reference set descriptor;
- the column order can be checked against the reference set descriptor.

Data structure

The Reference set descriptor reference set is a Component - Component - Integer reference set that specifies the structure of reference sets. Its structure is shown in the following table.

Table 5.2.11-4: Reference set descriptor reference set - Data structure

Field	Data type	Purpose	Mutable	Part of Primary Key
id	UUID	A 128 bit unsigned Integer, uniquely identifying this reference set member. Different versions of a *reference set member* share the same id but have different effectiveTime. This allows a *reference set member* to be modified or made inactive (i.e. removed from the active set) at a specified time.	NO	YES (Full / Snapshot)

effectiveTime	Time	The inclusive date or time at which this version of the identified reference set member became the current version. **Note** : In distribution files the effectiveTime should follow the short ISO date format (*YYYYMMDD*) and should not include the hours, minutes, seconds or timezone indicator. The current version of this reference set member at time *T* is the version with the most recent effectiveTime prior to or equal to time *T* .	YES	YES (Full) Optional (Snapshot)
active	Boolean	The state of the identified reference set member as at the specified effectiveTime . If active = 1 (true) the reference set member is part of the current version of the set, if active = 0 (false) the reference set member is not part of the current version of the set.	YES	NO
moduleId	SCTID	Identifies the SNOMED CT module that contains this reference set member as at the specified effectiveTime . The value must be a subtype of 900000000000443000 | Module (core metadata concept) | within the metadata hierarchy .	YES	NO
refsetId	SCTID	Identifies the reference set to which this reference set member belongs. In this case, the refsetId is always 900000000000456007 | Reference set descriptor | as there is only one reference set of this type.	NO	NO
referencedComponentId	SCTID	A reference to the SNOMED CT component to be included in the reference set . Refers to the concept that identifies the reference set (or reference set type) defined by this descriptor.	NO	NO
attributeDescription	SCTID	Specifies the name of an attribute that is used in the reference set to which this descriptor applies. Set to a descendant of 900000000000457003 | Reference set attribute (foundation metadata concept) | in the metadata hierarchy , that describes the additional attribute extending the reference set .	NO	NO
attributeType	SCTID	Specifies the data type of this attribute in the reference set to which this descriptor applies. Set to a descendant of 900000000000459000 | attribute type (foundation metadata concept) | in the metadata hierarchy , that describes the type of the additional attribute extending the reference set .	NO	NO
attributeOrder	integer	Specifies the position of this attribute in the reference set to which this descriptor applies. A zero value identifies the referencedComponentId within the reference set . Other values specify an additional attributes by its position relative to the referencedComponentId . Within a particular descriptor, attributeOrder values for a particular referencedComponentId must be contiguous. An unsigned Integer , providing an ordering for the additional attributes extending the reference set .	NO	NO

At least one row must exist for each reference set included in a release. This row must have an attributeOrder value of '0' and an attributeType of ' component type' (or one of its descendants). The referencedComponentId identifies the reference set defined by the descriptor.

There is one additional row for each additional column present in the specified reference set .

Creation of Reference set descriptor data is mandatory when creating a new reference set in the International Release or in a National Extension .

Creation of a Reference set descriptor is optional when creating a reference set in another Extension. If a descriptor is not created, the descriptor of the closest ancestor of the reference set is used when validating reference set member records.

Metadata

The following metadata in the |Foundation metadata concept | hierarchy supports the reference set descriptor reference set .

The Reference Set Descriptor Reference Set is specified by the 900000000000456007 |Reference set descriptor| concept in the metadata hierarchy.

- 900000000000441003 |SNOMED CT Model Component|
    - 900000000000454005 |Foundation metadata concept|
        - 900000000000455006 |Reference set|
            - 900000000000456007 |Reference set descriptor|

Table 5.2.11-4: Reference Set Descriptor Concept in the Metadata Hierarchy

Values in the Reference Set are populated from:

- 900000000000454005 |Foundation metadata concept|
    - 900000000000457003 |Reference set attribute|
        - 900000000000458008 |Attribute description|
        - 900000000000459000 |Attribute type|
            - 900000000000460005 |Component type|
                - 900000000000461009 |Concept type component|
                - 900000000000462002 |Description type component|
                - 900000000000463007 |Relationship type component|
                - 900000000000464001 |Reference set member type component|
            - 900000000000465000 |String|
                - 900000000000466004 |Text|
                    - 900000000000467008 |Single character|
                    - 900000000000468003 |Text < 256 bytes|
                - 900000000000469006 |URL|
                    - 900000000000470007 |HTML reference|
                    - 900000000000471006 |Image reference| ...
                - 900000000000474003 |UUID|
                - 900000000000475002 |Time|
            - 900000000000476001 |Integer|
                - 900000000000477005 |Signed integer|
                - 900000000000478000 |Unsigned integer|
            - 900000000000460005 |Component type| ...
            - 900000000000465000 |String| ...
            - 900000000000476001 |Integer| ...
        - 900000000000479008 |Attribute order|
        - 900000000000491004 |Attribute value| ...

Table 5.2.11-4: Reference Set Attribute Metadata Hierarchy

Reference Set Descriptor and Example Data

Notes on the tables used to show descriptors and examples
The reference set example tables on this page have been revised as follows to aid clarity and understanding:

- The first four columns which are present in all release files are not shown. The omitted columns (id, effectiveTime, active, moduleId) are used in the same way in all referenced sets to support identification, versioning and packaging. They do not directly affect the specific features of a particular reference set or reference set type.
- Reference set columns that contain SNOMED CT identifiers are expanded to show details of the concept or description referenced by that identifier. In some cases, the term is shown in the same column using the expression syntax, in other cases an additional column with a name suffix '_term' has been added. In the standard reference set files only the identifier is present in the column and there is no added column for the term. When using reference sets, the term and other details of the component are looked up from the relevant component release files.

Descriptor Template

The table below shows the descriptor that defines the structure of the 900000000000456007 |Reference set descriptor|. Note that this descriptor is itself part of the 900000000000456007 |Reference set descriptor| that it describes!

Table 5.2.11-4: Refset Descriptor rows for

refsetId	referencedComponentId (Referenced component)	attributeDescription (Attribute description)	attributeType (Attribute type)	attributeOrder (Attribute order)
900000000000456007 |Reference set descriptor|	900000000000456007 |Reference set descriptor|	449608002 |Referenced component|	900000000000461009 |Concept type component |	0
900000000000456007 |Reference set descriptor|	900000000000456007 |Reference set descriptor|	900000000000458008 |Attribute description|	900000000000461009 |Concept type component |	1
900000000000456007 |Reference set descriptor|	900000000000456007 |Reference set descriptor|	900000000000459000 |Attribute type|	900000000000461009 |Concept type component |	2
900000000000456007 |Reference set descriptor|	900000000000456007 |Reference set descriptor|	900000000000479008 |Attribute order|	900000000000478000 |Unsigned integer|	3

5.2.12 Module Dependency Reference Set

Purpose

The |Module dependency reference set| represents dependencies between different SNOMED CT release modules. In each case, the dependency indicates which targetEffectiveTime of each particular module a given sourceEffectiveTime of the dependent module requires.

Data structure

The 900000000000534007 |Module dependency reference set| is a String - String reference set which is used to represent dependencies between modules, taking account of module versioning. Its structure is shown in the following table.

Table 5.2.12-4: Module dependency reference set - Data Structure

Field	**Data type**	**Purpose**	Mutable	**Part of Primary Key**
id	UUID	A 128 bit unsigned Integer, uniquely identifying this reference set member. Different versions of a *reference set member* share the same id but have different effectiveTime. This allows a *reference set member* to be modified or made inactive (i.e. removed from the active set) at a specified time.	NO	YES (Full / Snapshot)
effectiveTime	Time	The inclusive date or time at which this version of the identified reference set member became the current version. **Note**: In distribution files the effectiveTime should follow the short ISO date format (*YYYYMMDD*) and should not include the hours, minutes, seconds or timezone indicator. The current version of this reference set member at time *T* is the version with the most recent effectiveTime prior to or equal to time *T*.	YES	YES (Full) Optional (Snapshot)
active	Boolean	The state of the identified reference set member as at the specified effectiveTime. If active = 1 (true) the reference set member is part of the current version of the set, if active = 0 (false) the reference set member is not part of the current version of the set. **Note**: *A module dependency should only be inactivated if it is found to be erroneous.*	YES	NO
moduleId	SCTID	Identifies the SNOMED CT module that contains this reference set member as at the specified effectiveTime. The value must be a subtype of 900000000000443000 |Module (core metadata concept)| within the metadata hierarchy. The moduleId for in the Module Dependency Reference Set represents the source module (i.e. the module declaring a dependency on another module). **Note**: In all other situations moduleId is mutable. However, in the Module Dependency Reference Set a change to the moduleId would also change the source of the dependency. Therefore, it should **not** be treated as mutable.	YES	NO
refsetId	SCTID	Identifies the reference set to which this reference set member belongs. In this case, always 900000000000534007 |Module dependency reference set| as there is only one Module Dependency Reference set.	NO	NO
referencedComponentId	SCTID	A reference to the SNOMED CT component to be included in the reference set. This refers to the target of the dependency (i.e. the module on which the module identified by moduleId depends).	NO	NO
sourceEffectiveTime	Time	The effective time of the dependent source module (identified by moduleId). This specifies a version of that module, consisting of all components that have the same moduleId as this refset member in their states as at the specified targetEffectiveTime.	YES	NO

targetEffectiveTime	Time	The effective time of the target module required to satisfy the dependency (identified by referencedComponentId). This specifies a version of that module , consisting of all components with the moduleId specified by referencedComponentId in their states as at the specified targetEffectiveTime .	YES	NO

Rules and Guidance

Introduction to Modules

Each row in the release files for components and reference set members has a moduleId. This refers to the module that the component is maintained in. Each module is either part of the SNOMED International Release or part of a single SNOMED CT Extension . The moduleId has a partition-id which indicates whether it is part of the SNOMED International Release and, if not, its namespace identifier indicates the SNOMED CT Extension that it belongs to.

A module is simply a collection of SNOMED CT components maintained as a unit by a single organization. It is the organization 's responsibility to organize the components in each extension that it is responsible for into one or more modules, in a way that best fits its business needs.

A module is represented by a descendant of 900000000000443000 |Module| in the metadata hierarchy. The immediate subtype descendants of 900000000000443000 |Module| represent groups of modules maintained by an organization and subtypes of these can be used to arrange that organizations modules into a number of groups. For example, all modules maintained by SNOMED International will be children of 900000000000445007 |SNOMED International maintained module|.

At any point in time a component must be in one, and only one module. It is possible for components and reference set members to be moved between modules (subject to constraints explained elsewhere). In this case, a new row is added to the release file with the same id but with a new effectiveTime and a new moduleId.

Introduction to Module Dependencies

Each extension must include one or more modules. Each module must be part of either the SNOMED International Release or one and only one extension. A module may not move from one extension to another over time. If components or reference set members in a module are to be moved from an extension to the SNOMED International Release or to another extension, they must either be added to an existing or newly created module maintained by the destination organization.

The 900000000000443000 |Module| sub-hierarchy does NOT represent dependencies between module. Instead, module dependencies are modeled using the 900000000000534007 |Module dependency reference set| .

At the point of release, if any component within a module has changed, then a new row must be added to the |Module dependency reference set| for each dependency of that module. The effectiveTime of the added rows must set to the date of the new release. The updated |Module dependency reference set| records indicate that some components within the module have been updated in this release. If there have been no additions, updates or inactivations of components or reference set members within a module, then a new |Module dependency reference set| records need not be added unless there is a requirement to declare that the unchanged module is compatible with a later release of the module(s) on which it dependents.

Identifying and Versioning Module Dependencies

id

The recommended practice is for the refset.id column to contain the same identifier for all versions of the dependencies between the same pair of modules. This approach means that at any given time only one version of each module has effective dependencies. The dependencies of earlier versions can be reviewed by reviewing a snapshot for the effectiveTime of the earlier release.

Value of the id column
An alternative approach has been suggested by some people in which a new identifier is allocated to each dependency of each module. This would then mean that all past dependencies would be visible in a snapshot view. It would also mean that it would be possible release updated dependencies for an existing module version while also releasing more up-to-date versions of the same module with different dependencies. This added flexibility comes at the price of additional complexity and for the time-being the International Release modules continue to use the simpler approach in which each new version of a dependency supersedes the dependency between earlier versions of the same pair of modules.

effectiveTime

The effectiveTime of at least one row for each pair of modules should be the same as the sourceEffectiveTime. Otherwise, there will be a period of time when a snapshot view will not show the dependencies. However, it is theoretically possible for an additional row to be added with a later effectiveTime in cases where an otherwise unchanged release of an extension, declares itself to be compatible with an updated release of the target module (in this case the effectiveTime and targetEffectiveTime are changed but the sourceEffectiveTime remains unchanged.

active

A module dependency only needs to be inactivated if the dependency is found to be erroneous. This is because, the module dependency is specific to a particular version of the source and target module. Therefore, if that dependency was valid at the outset it remains valid indefinitely in respect of those specified module versions, even if the dependencies between subsequent versions differ.

refsetId

Module version dependencies are represented using a single 900000000000534007 |Module dependency reference set|. Thus all module dependency rows have the same refsetId (900000000000534007 |Module dependency reference set (foundation metadata concept)|).

It is the responsibility of the organization owning and maintaining a dependent module to identify all modules on which it depends. They do this by adding rows to the 900000000000534007 |Module dependency reference set| within the dependent module. Because these added member must be in the dependent module, the moduleId of the reference set member record is also the identifier of the dependent (source) module.

Module Identification

Source Module (moduleId)

The moduleId column not only indicates that this reference set member is in the specified module, it also indicates that this is the module that is the source of the dependency. As a result, in this reference set the moduleId column is immutable (i.e Mutable=NO). This is an exception to the usual rule and implies that a member of this reference set cannot move from one module to another.

Target Module (referencedComponentId)

The target module on which the source module depends is identified by the referencedComponentId. Like the source module this is also immutable and this implies that if a module ceases to be dependent on another module, a new row inactivating the dependency can be added but the same member cannot be used to represent a different dependency (even if that dependency is a direct replacement of the inactivated dependency). However as noted above,

A module version may depend on one or more other module versions, and many module versions may have a dependency on a single module version. Cyclic module version dependencies are not allowed. If module-A depends on module-B, then module-B cannot depend on module-A.

Dependencies are not transitive and this means that dependencies cannot be inferred from a chain of dependencies. If module-A depends on module-B and module-B depends on module-C, the dependency of module-A on module-C must still be stated explicitly.

Any release should consist of a set of module versions that are certified as being compatible. Each release should also identify other module versions that it is dependent on even when these are outside the scope of the release. For example, the dependencies of modules in an Extension on the International Release must be stated.

Dependencies are specified between module versions, not just dependencies between modules. Therefore, it is possible to specify a dependency from a module released on one date to an earlier version of another module. The version of the dependent module is specified by the sourceEffectiveTime and the version of the module on which it depends is specified by the targetEffectiveTime.

Metadata

The following metadata in the "Foundation metadata concept " hierarchy supports this reference set :

Table 5.2.12-4: Module Dependency Reference Set in the Metadata Hierarchy

```
900000000000454005 |Foundation metadata concept|
    900000000000455006 |Reference set|
        900000000000534007 |Module dependency|
```

Reference Set Descriptor and Example Data

Notes on the tables used to show descriptors and examples
The reference set example tables on this page have been revised as follows to aid clarity and understanding:

- The first four columns which are present in all release files are not shown. The omitted columns (id, effectiveTime, active, moduleId) are used in the same way in all referenced sets to support identification, versioning and packaging. They do not directly affect the specific features of a particular reference set or reference set type.
- Reference set columns that contain SNOMED CT identifiers are expanded to show details of the concept or description referenced by that identifier. In some cases, the term is shown in the same column using the expression syntax, in other cases an additional column with a name suffix '_term' has been added. In the standard reference set files only the identifier is present in the column and there is no added column for the term. When using reference sets, the term and other details of the component are looked up from the relevant component release files.

Descriptor

The table below shows the descriptor that defines the structure of the 900000000000534007 |Module dependency reference set| .

Table 5.2.12-4: Refset Descriptor rows for the Module Dependency

refsetId	referencedComponentId (Referenced component)	attributeDescription (Attribute description)	attributeType (Attribute type)	attributeOrder (Attribute order)	
900000000000456007 |Reference set descriptor|	900000000000534007 |Module dependency|	900000000000535008 |Dependency target|	900000000000461009 |Concept type component|	0	

refsetId	referencedComponentId (Referenced component)	attributeDescription (Attribute description)	attributeType (Attribute type)	attributeOrder (Attribute order)	
900000000000456007 |Reference set descriptor|	900000000000534007 |Module dependency|	900000000000536009 |Source effective time|	900000000000475002 |Time|	1	
900000000000456007 |Reference set descriptor|	900000000000534007 |Module dependency|	900000000000537000 |Target effective time|	900000000000475002 |Time|	2	

Note: The table above omits the initial four columns of data present in the release file. These follow the standards versioning pattern id, effectiveTime, active, active. Additionally, to aid understanding, the table above also shows the term from one of the descriptions associated with each of the identified concept. The release file only contains the identifier.

Example Data

Example The table below holds example entries for the 900000000000534007 |Module dependency reference set| in a snapshot view of the January 2014 SNOMED CT International Release .

This SNOMED CT International Release contains three modules:

- 900000000000012004 |SNOMED CT model component| which has no dependencies;
- 900000000000207008 |SNOMED CT core| which depends on the 900000000000012004 |SNOMED CT model component| ; and
- 449080006 |SNOMED CT to ICD-10 rule-based mapping module| which depends on both the other modules.

In this case all the 2014-01-31 modules depend on 2014-01-31 versions of the other modules. However, in some case a module may depend on an earlier version of another model (e.g. an extension module may be releases after the SNOMED CT International Release to which it applies).

Dependencies are not transitive. The fact that 449080006 |SNOMED CT to ICD-10 rule-based mapping module| is dependent on 900000000000207008 |SNOMED CT core| may seem to imply a dependency on 900000000000012004 |SNOMED CT model component| . However, in practice all dependencies must be explicitly specified, not just immediate dependencies.

Table 5.2.12-4: Sample content from

moduleId	refsetId	referencedComponentId (Dependency target)	sourceEffectiveTime (Source effective time)	targetEffectiveTime (Target effective time)
900000000000207008 |SNOMED CT core|	900000000000534007 |Module dependency|	900000000000012004 |SNOMED CT model component|	20140131	20140131
449080006 |SNOMED CT to ICD-10 rule-based mapping module|	900000000000534007 |Module dependency|	900000000000012004 |SNOMED CT model component|	20140131	20140131
449080006 |SNOMED CT to ICD-10 rule-based mapping module|	900000000000534007 |Module dependency|	900000000000207008 |SNOMED CT core|	20140131	20140131

5.2.13 Description Format Reference Set

Purpose

The 900000000000538005 |Description format reference set|specifies the text format and maximum length of each supported description type. This permits additional description types to be specified in future in addition to the three existing description types (synonym, fully specified name and textual definition).

Data structure

The 900000000000538005 |Description format reference set|is a C-I (component- Integer) reference set which is used to specify the length and format of the terms in descriptions of this description type. Its structure is shown in the following table.

Table 5.2.13-4: Description format reference set - Data structure

Field	**Data type**	**Purpose**	Mutable	**Part of Primary Key**
id	UUID	A 128 bit unsigned Integer, uniquely identifying this reference set member. Different versions of a *reference set member* share the same id but have different effectiveTime. This allows a *reference set member* to be modified or made inactive (i.e. removed from the active set) at a specified time.	NO	YES (Full / Snapshot)
effectiveTime	Time	The inclusive date or time at which this version of the identified reference set member became the current version. **Note** : In distribution files the effectiveTime should follow the short ISO date format (*YYYYMMDD*) and should not include the hours, minutes, seconds or timezone indicator. The current version of this reference set member at time *T* is the version with the most recent effectiveTime prior to or equal to time *T* .	YES	YES (Full) Optional (Snapshot)
active	Boolean	The state of the identified reference set member as at the specified effectiveTime . If active = 1 (true) the reference set member is part of the current version of the set, if active = 0 (false) the reference set member is not part of the current version of the set.	YES	NO
moduleId	SCTID	Identifies the SNOMED CT module that contains this reference set member as at the specified effectiveTime . The value must be a subtype of 900000000000443000 | Module (core metadata concept) | within the metadata hierarchy.	YES	NO
refsetId	SCTID	Identifies the reference set to which this reference set member belongs. In this case, a subtype descendant of: 900000000000538005 | Description format reference set (foundation metadata concept) |	NO	NO
referencedComponentId	SCTID	A reference to the SNOMED CT component to be included in the reference set. Refers to the concept that represents the description type for which this row defines the format.	NO	NO

descriptionFormat	SCTID	A reference to a subtype of 900000000000539002 |Description format (foundation metadata concept)| attribute which specifies the format of terms in descriptions of this description type.	NO	NO
descriptionLength	integer	The maximum length in bytes of the terms in descriptions of this description type.	NO	NO

Metadata

The following metadata supports the description format reference set:

Table 5.2.13-4: Description Format Reference Set in the Metadata Hierarchy

```
900000000000454005 |Foundation metadata concept|
    900000000000455006 |Reference set|
        900000000000538005 |Description format|
```

Reference Set Descriptor and Example Data

Notes on the tables used to show descriptors and examples
The reference set example tables on this page have been revised as follows to aid clarity and understanding:

- The first four columns which are present in all release files are not shown. The omitted columns (id, effectiveTime, active, moduleId) are used in the same way in all referenced sets to support identification, versioning and packaging. They do not directly affect the specific features of a particular reference set or reference set type.
- Reference set columns that contain SNOMED CT identifiers are expanded to show details of the concept or description referenced by that identifier. In some cases, the term is shown in the same column using the expression syntax, in other cases an additional column with a name suffix '_term' has been added. In the standard reference set files only the identifier is present in the column and there is no added column for the term. When using reference sets, the term and other details of the component are looked up from the relevant component release files.

Descriptor

The table below shows the descriptor that defines the structure of the 900000000000538005 |Description format reference set|.

Table 5.2.13-4: Refset Descriptor rows for Description Format Reference Set

refsetId	referencedComponentId (Referenced component)	attributeDescription (Attribute description)	attributeType (Attribute type)	attributeOrder (Attribute order)
900000000000456007 |Reference set descriptor|	900000000000538005 |Description format|	900000000000462002 |Description type component|	900000000000461009 |Concept type component|	0
900000000000456007 |Reference set descriptor|	900000000000538005 |Description format|	900000000000539002 |Description format|	900000000000461009 |Concept type component|	1

refsetId	referencedComponentId (Referenced component)	attributeDescription (Attribute description)	attributeType (Attribute type)	attributeOrder (Attribute order)
900000000000456007 |Reference set descriptor|	900000000000538005 |Description format|	900000000000544009 |Description length|	900000000000478000 |Unsigned integer|	2

Example Data

This example holds the all the members of the 900000000000538005 |Description format reference set| in the SNOMED CT International Release for July 2014. Other members may added to future versions of the International Release if new description types are introduced. Owners of Extensions that support additional description types must also add members to the 900000000000538005 |Description format reference set|.

Table 5.2.13-4: Sample Content from the Description Format Reference Set

refsetId	referencedComponentId (Description type component)	descriptionFormat (Description format)	descriptionLength (Description length)
900000000000538005 |Description format|	900000000000003001 |Fully specified name|	900000000000540000 |Plain text|	255
900000000000538005 |Description format|	900000000000013009 |Synonym|	900000000000540000 |Plain text|	255
900000000000538005 |Description format|	900000000000550004 |Definition|	900000000000540000 |Plain text|	4096

Note: The tables above omit the initial four columns of data present in the release file. These follow the standards versioning pattern id, effectiveTime, active, active. Additionally, to aid understanding, the tables above also show the term from one of the descriptions associated with each of the identified concept. The release file only contains the identifier.

5.2.14 Map Correlation and Origin Reference Set

Purpose

The |Map correlation and origin type reference set| is used to meet the requirements for representation of maps between codes in another code system (other-codes) and a SNOMED CT concept where the following requirements apply.

1. A requirement to indicate the degree of correlation between the SNOMED CT concept and the other-codes.
2. A requirement to indicate whether a concept or code was added to either code system as a result of the mapping process and, in this case, to indicate in which code system the concept or code originated.
3. A requirement to represent the SNOMED CT attribute to which the other-code should be applied in order to capture the full specificity of the value represented by the other-code.
4. No requirements for mapping rules or advice to be included with each map.

Data Structure

Table 5.2.14-1: Map Correlation and Origin Reference Set - Data Structure

Field	Data type	Purpose	Mutable	Part of Primary Key
id	UUID	A 128 bit unsigned Integer, uniquely identifying this reference set member. Different versions of a *reference set member* share the same id but have different effectiveTime. This allows a *reference set member* to be modified or made inactive (i.e. removed from the active set) at a specified time.	NO	YES (Full / Snapshot)
effectiveTime	Time	The inclusive date or time at which this version of the identified reference set member became the current version. **Note** : In distribution files the effectiveTime should follow the short ISO date format (*YYYYMM D D*) and should not include the hours, minutes, seconds or timezone indicator. The current version of this reference set member at time *T* is the version with the most recent effectiveTime prior to or equal to time *T* .	YES	YES (Full) Optional (Snapshot)
active	Boolean	The state of the identified reference set member as at the specified effectiveTime . If active = 1 (true) the reference set member is part of the current version of the set, if active = 0 (false) the reference set member is not part of the current version of the set.	YES	NO
moduleId	SCTID	Identifies the SNOMED CT module that contains this reference set member as at the specified effectiveTime . The value must be a subtype of 900000000000443000 | Module (core metadata concept) | within the metadata hierarchy.	YES	NO
refsetId	SCTID	Identifies the reference set to which this reference set member belongs. In this case, a subtype descendant of: | Map correlation and origin type reference set (foundation metadata concept) |	NO	NO
referencedComponentId	SCTID	A reference to the SNOMED CT component to be included in the reference set. The SNOMED CT concept that is associated with the code in the other terminology or code system.	NO	NO
mapTarget	String	The other-code to/from which the concept is mapped.	NO	NO
attributeId	SCTID	A reference to the SNOMED CT concept representing the attribute to which the referencedComponentId (other-code) applies. In some cases, other-codes may be overloaded with a meaning that combines the meaning of a specific attribute with a value applied to it in the SNOMED CT concept model[1] , in these cases accurate mapping needs to specify both aspects of the meaning. The attributeId provides effective disambiguation in these cases. Values of attributeId are restricted to subtypes of | Concept model attribute | .	YES	NO
correlationId	SCTID	The correlation between the SNOMED CT concept and the other-code. Possible values are the following subtypes of 447247004 | SNOMED CT source code to target map code correlation value | : 447559001 | Broad to narrow map from SNOMED CT source code to target code | 447557004 | Exact match map from SNOMED CT source code to target code | 447558009 | Narrow to broad map from SNOMED CT source code to target code | 447560006 | Partial overlap between SNOMED CT source code and target code |	YES	NO

contentOriginId	SCTID	Indication of whether the concept was initially in one of the terminologies (SNOMED CT or other-codes) and added to the other as part of mapping or was in both terminologies at the outset. Values are subtypes of 705116007 | Original code system source for linked content value | .	YES	NO

Related Links

- For further information see Using LOINC with SNOMED CT: 4.2.1 LOINC Part Map Reference Set.

5.2.15 Code to Expression Reference Set

Purpose

The |Code to expression type reference set| is designed to enable associations between codes in another code system (other-codes) and SNOMED CT concepts, where the following constraints apply:

1. Some of the other-codes cannot be mapped to an individual SNOMED CT concept.
2. Licensing conditions (or other considerations) prevent addition of new SNOMED CT concepts to represent the same meaning as the other-codes.
3. The other-codes can be logically defined using the SNOMED concept model to represent the same meaning (sufficiently defined) or a similar though less specific meaning (primitive).
4. Other requirements similar for those applicable to mapping may also apply including:
    a. An indication of the degree of correlation between the other-code and the SNOMED CT expression.
    b. An indication of whether the other-code was created before any single concept representation of that meaning in SNOMED CT or whether the single concept representation in SNOMED CT predated the creation of the association.

Data Structure

The general approach to the above requirements is to associate each of the other-codes with a representation of the same logic based definition as would have been applied to a SNOMED CT concept with that meaning. However, since the other-code are not identified by an SCTID, the logical definition cannot be represented using defining relationships. There are two potential approaches to this, one would be to use a general purpose description logic language (e.g. OWL) and the other is to use a SNOMED CT expression to represent each definition. The |Code to expression type reference set| is designed to support the expression-based approach.

Table 5.2.15-1: Code to expression type reference set - Data Structure

Field	**Data type**	**Purpose**	Mutable	**Part of Primary Key**
id	UUID	A 128 bit unsigned Integer, uniquely identifying this reference set member. Different versions of a *reference set member* share the same id but have different effectiveTime. This allows a *reference set member* to be modified or made inactive (i.e. removed from the active set) at a specified time.	NO	YES (Full / Snapshot)
effectiveTime	Time	The inclusive date or time at which this version of the identified reference set member became the current version. **Note** : In distribution files the effectiveTime should follow the short ISO date format (*YYYYMM DD*) and should not include the hours, minutes, seconds or timezone indicator. The current version of this reference set member at time *T* is the version with the most recent effectiveTime prior to or equal to time *T* .	YES	YES (Full) Optional (Snapsho t)

active	Boolean	The state of the identified reference set member as at the specified effectiveTime . If active = 1 (true) the reference set member is part of the current version of the set, if active = 0 (false) the reference set member is not part of the current version of the set.	YES	NO
moduleId	SCTID	Identifies the SNOMED CT module that contains this reference set member as at the specified effectiveTime . The value must be a subtype of 900000000000443000 | Module (core metadata concept) | within the metadata hierarchy .	YES	NO
refsetId	SCTID	Identifies the reference set to which this reference set member belongs. In this case, a subtype descendant of: | Code to expression type reference set |	NO	NO
referencedComponentId	SCTID	A reference to the SNOMED CT component to be included in the reference set . A subtype of 705113004 | Terminology system | identifying the code system from which the code in the mapTarget field is derived. • For example: 705114005 | LOINC Code System | .	NO	NO
mapTarget	String	The other-code to/from which the concept is mapped.	NO	NO
expression	String	A SNOMED CT expression that represents the SNOMED CT definition of the other-code . This expression may be a stated or inferred view of the definition provided that documentation of each identified reference set specifies the view provided. The expression must conform to the syntax defined in the Error rendering macro 'sp-plaintextbody-link' Conversion context did not contain original content entity. (http://snomed.org/scg).	YES	NO
definitionStatusId	SCTID	Indicates whether or not the expression contains a sufficient definition of the other-code in the mapTarget field. Possible values are the following subtypes of 900000000000444006 | Definition status | : 900000000000074008 | Necessary but not sufficient concept definition status | 900000000000073002 | Sufficiently defined concept definition status |	YES	NO
correlationId	SCTID	The correlation between the SNOMED CT expression and the other-code . Possible values are the following subtypes of 447247004 | SNOMED CT source code to target map code correlation value | : 447559001 | Broad to narrow map from SNOMED CT source code to target code | 447557004 | Exact match map from SNOMED CT source code to target code | 447558009 | Narrow to broad map from SNOMED CT source code to target code | 447560006 | Partial overlap between SNOMED CT source code and target code | When these values are applied to this reference set type, the phrase " *SNOMED source code* " is interpreted as meaning " *SNOMED expression* " and "target code" refers to the other-code . .	YES	NO
contentOriginId	SCTID	Indication of whether the concept was initially in one of the terminologies (SNOMED CT or other-codes) and added to the other as part of mapping or was in both terminologies at the outset. Values are subtypes of 705116007 | Original code system source for linked content value | .	YES	NO

refs

Related Links

- For further information see Using LOINC with SNOMED CT: 4.2.2 LOINC Term to Expression Reference Set.

5.2.16 MRCM Domain Reference Set

Purpose

An 723589008 |MRCM domain reference set| enumerates the concept domains to which SNOMED CT attributes may be applied, and provides additional information to support these concept domains.

Each concept domain is uniquely identified by a SNOMED CT concept. When the scope of a domain covers the concepts in a particular hierarchy (or subhierarchy), the supertype concept of this hierarchy (or subhierarchy) is used to identify the domain. When a domain is defined based on membership in a reference set, the associated reference set concept is used to identify the domain. In some situations, a query may be required to define a complex domain. In these cases, the query's expansion reference set (referred to by the 'referencedComponent' of the relevant Query reference set) is used to identify the domain.

For each domain in the SNOMED CT concept model, the 723589008 |MRCM domain reference set| will contain exactly one member. This reference set member will include an Expression Constraint that defines the concepts in the domain, the identifier of the immediate parent domain (or domains), the domain constraint defined in terms of its proximal primitive concepts and associated mandatory refinements, a generic Domain Expression Template for both precoordinated and postcoordinated content, and a reference to the associated guidance that provides additional human-readable text describing this domain. Please note that it is anticipated that the generic Domain Expression Templates will be specialized further for authoring of specific subdomains using specializations stored in a Template Library.

Data Structure

An 723589008 |MRCM domain reference set| is structured as shown in the following table.

Field	**Data type**	**Purpose**	Mutable	**Part of Primary Key**
id	UUID	A 128 bit unsigned Integer, uniquely identifying this reference set member. Different versions of a *reference set member* share the same id but have different effectiveTime. This allows a *reference set member* to be modified or made inactive (i.e. removed from the active set) at a specified time.	NO	YES (Full / Snapshot)
effectiveTime	Time	The inclusive date or time at which this version of the identified reference set member became the current version. **Note** : In distribution files the effectiveTime should follow the short ISO date format (*YYYYMMDD*) and should not include the hours, minutes, seconds or timezone indicator. The current version of this reference set member at time *T* is the version with the most recent effectiveTime prior to or equal to time *T* .	YES	YES (Full) Optional (Snapshot)
active	Boolean	The state of the identified reference set member as at the specified effectiveTime . If active = 1 (true) the reference set member is part of the current version of the set, if active = 0 (false) the reference set member is not part of the current version of the set.	YES	NO

moduleId	SCTID	Identifies the SNOMED CT module that contains this reference set member as at the specified effectiveTime . The value must be a subtype of 900000000000443000 | Module (core metadata concept) | within the metadata hierarchy.	YES	NO
refsetId	SCTID	Identifies the reference set to which this reference set member belongs. In this case, a subtype descendant of: 723589008 | MRCM domain reference set |	NO	NO
referencedComponentId	SCTID	A reference to the SNOMED CT component to be included in the reference set. A reference to the SNOMED CT concept that identifies the relevant concept domain.	NO	NO
domainConstraint	String	An expression constraint, which defines the set of concepts included in the given concept domain. This string can be parsed using the ABNF syntax defined for the Expression Constraint Language.	YES	NO
parentDomain	String	An expression constraint, which represents the set of immediate parent domains. An immediate parent domain is a domain that is a proper superset of the given domain, and which is not a proper superset of any other parent domain.	YES	NO
proximalPrimitive Constraint	String	The domain constraint, as it would be represented for proximal primitive modelling. If the domain concept is sufficiently defined, then its proximal primitive parent will be used instead, while if the domain concept is primitive, then the concept itself is used. Additional constraints on the proximal primitive parent are also included. The expansion of the given constraint must be further filtered to find those concepts with a definitionStatusId = 900000000000074008 | Primitive | . This string can be parsed using the ABNF syntax defined for the Expression Constraint Language.	YES	NO
proximalPrimitive Refinement	String	The template representation of any additional refinements that are required to model in the given domain using proximal primitive modelling. These mandatory refinements reflect the defining relationships of the domain concept, when it is sufficiently defined. This string can be parsed using the 'refinement' rule in the ABNF syntax defined for the Expression Constraint Language .	YES	NO
domainTemplate ForPrecoordination	String	A general template that may be used to author precoordinated content. This template incorporates all of the mandatory attribute domain and range rules rules for precoordinated SNOMED CT content. This string can be parsed using the Expression Template Language (currently under development).	YES	NO
domainTemplate ForPostcoordination	String	A general template that may be used to author postcoordinated content. This template incorporates all of the mandatory attribute domain and range rules rules for postcoordinated SNOMED CT content. This string can be parsed using the Expression Template Language (currently under development).	YES	NO
guideURL	URL	A Uniform Resource Locator (URL) that references a web resource in which the given domain is described in further detail. This URL uses the following pattern: " http://snomed.org/dom *<conceptId>* "	YES	NO

Metadata

The following metadata hierarchy supports this reference set:

- 900000000000454005 |Foundation metadata concept|
    - 900000000000455006 |Reference set|
        - 723564002 |MRCM reference set|
            - 723589008 |MRCM domain reference set|
                - 723560006 |MRCM domain international reference set|
    - 900000000000457003 |Reference set attribute|
        - 723565001 |Domain constraint|
        - 723570008 |Guide URL|
        - 723566000 |Parent domain|
        - 723567009 |Proximal primitive constraint|
        - 723568004 |Proximal primitive refinement|
        - 723569007 |Template|
            - 723599003 |Domain template|
                - 723600000 |Domain template for precoordination|
                - 723601001 |Domain template for postcoordination|

Reference Set Descriptor and Example Data

Notes on the tables used to show descriptors and examples
The reference set example tables on this page have been revised as follows to aid clarity and understanding:

- The first four columns which are present in all release files are not shown. The omitted columns (id, effectiveTime, active, moduleId) are used in the same way in all referenced sets to support identification, versioning and packaging. They do not directly affect the specific features of a particular reference set or reference set type.
- Reference set columns that contain SNOMED CT identifiers are expanded to show details of the concept or description referenced by that identifier. In some cases, the term is shown in the same column using the expression syntax, in other cases an additional column with a name suffix '_term' has been added. In the standard reference set files only the identifier is present in the column and there is no added column for the term. When using reference sets, the term and other details of the component are looked up from the relevant component release files.

Descriptor Template

The table below shows the reference set descriptor for a reference set that follows the 723589008 |MRCM domain reference set| pattern.

refsetId	referencedComponentId	attributeDescription	attributeType	attribute Order
900000000000456007 |Reference set descriptor|	723589008 |MRCM domain reference set|	449608002 |Referenced component|	900000000000461009 |Concept type component|	0
900000000000456007 |Reference set descriptor|	723589008 |MRCM domain reference set|	723565001 |Domain constraint|	707000009 |SNOMED CT parsable string|	1
900000000000456007 |Reference set descriptor|	723589008 |MRCM domain reference set|	723566000 |Parent domain|	707000009 |SNOMED CT parsable string|	2

refsetId	referencedComponentId	attributeDescription	attributeType	attribute Order
900000000000456007 | Reference set descriptor |	723589008 | MRCM domain reference set |	723567009 | Proximal primitive constraint |	707000009 | SNOMED CT parsable string |	3
900000000000456007 | Reference set descriptor |	723589008 | MRCM domain reference set |	723568004 | Proximal primitive refinement |	707000009 | SNOMED CT parsable string |	4
900000000000456007 | Reference set descriptor |	723589008 | MRCM domain reference set |	723600000 | Domain template for precoordination |	707000009 | SNOMED CT parsable string |	5
900000000000456007 | Reference set descriptor |	723589008 | MRCM domain reference set |	723601001 | Domain template for postcoordination |	707000009 | SNOMED CT parsable string |	6
900000000000456007 | Reference set descriptor |	723589008 | MRCM domain reference set |	723570008 | Guide URL |	707000009 | SNOMED CT parsable string |	7

Example Data

The table below shows some example rows from a reference set that uses the format of the 723589008 |MRCM domain reference set| .

Please note that the generic domain templates defined for the SNOMED CT International Edition are designed to support a proximal primitive parent authoring approach. However, domain templates included in an extension's 723589008 |MRCM domain reference set| may be designed to support a proximal parent authoring approach if required.

refsetId	referenced Component Id	domain Constraint	parent Domain	proximal Primitive Constraint	proximal Primitive Refinement	domainTemplateForPrecoordination	domainTemplateForPostcoordination	guideURL
723560006 | MRCM domain international reference set |	71388002 | Procedure (procedure) |	<< 71388002 | Procedure (procedure) |		<< 71388002 | Procedure (procedure) |		[[+id(<< 71388002 | Procedure (procedure) |)]]: [[0..*]] { [[0..*]] 260507000 | Access | = [[+id(<< 309795001 | Surgical access values (qualifier value) |)]], [[0..*]] 363699004 | Direct device | = [[+id(<< 49062001 | Device (physical object) |)]], [[0..*]] 363700003 | Direct morphology | = [[+id(<< 49755003 | Morphologically abnormal structure (morphologic abnormality) |)]], [[0..*]] 363701004 | Direct substance | = [[+id(<< 105590001 | Substance (substance) | OR << 373873005 | Pharmaceutical / biologic product (product) |)]], [[0..*]] 363702006 | Has focus | = [[+id(<< 404684003 | Clinical finding (finding) | OR << 71388002 | Procedure (procedure) |)]], [[0..*]] 363703001 | Has intent | = [[+id(<< 363675004 | Intents (nature of procedure values) (qualifier value) |)]], [[0..*]] 363710007 | Indirect device | = [[+id(<< 49062001 | Device (physical object) |)]], [[0..*]] 363709002 | Indirect morphology | = [[+id(<< 49755003 | Morphologically abnormal structure (morphologic abnormality) |)]], [[0..*]] 260686004 | Method | = [[+id(<< 129264002 | Action (qualifier value) |)]], [[0..*]] 260870009 | Priority | = [[+id(<< 272125009 | Priorities (qualifier value) |)]], [[0..*]] 405815000 | Procedure device | = [[+id(<< 49062001 | Device (physical object) |)]], [[0..*]] 405816004 | Procedure morphology | = [[+id(<< 49755003 | Morphologically abnormal structure (morphologic abnormality) |)]], [[0..*]] 363704007 | Procedure site | = [[+id(<< 442083009 | Anatomical or acquired body structure (body structure) |)]], [[0..*]] 405813007 | Procedure site - Direct | = [[+id(<< 442083009 | Anatomical or acquired body structure (body structure) |)]], [[0..*]] 405814001 | Procedure site - Indirect | = [[+id(<< 442083009 | Anatomical or acquired body structure (body structure) |)]], [[0..*]] 370131001 | Recipient category | = [[+id(<< 125676002 | Person (person) | OR << 35359004 | Family (social concept) | OR << 133928008 |	[[+scg(<< 71388002 | Procedure (procedure) |)]]: [[0..*]] { [[0..*]] 260507000 | Access | = [[+scg(<< 309795001 | Surgical access values (qualifier value) |)]], [[0..*]] 363699004 | Direct device | = [[+scg(<< 49062001 | Device (physical object) |)]], [[0..*]] 363700003 | Direct morphology | = [[+scg(<< 49755003 | Morphologically abnormal structure (morphologic abnormality) |)]], [[0..*]] 363701004 | Direct substance | = [[+scg(<< 105590001 | Substance (substance) | OR << 373873005 | Pharmaceutical / biologic product (product) |)]], [[0..*]] 363702006 | Has focus | = [[+scg(<< 404684003 | Clinical finding (finding) | OR << 71388002 | Procedure (procedure) |)]], [[0..*]] 363703001 | Has intent | = [[+scg(<< 363675004 | Intents (nature of procedure values) (qualifier value) |)]], [[0..*]] 363710007 | Indirect device | = [[+scg(<< 49062001 | Device (physical object) |)]], [[0..*]] 363709002 | Indirect morphology | = [[+scg(<< 49755003 | Morphologically abnormal structure (morphologic abnormality) |)]], [[0..*]] 260686004 | Method | = [[+scg(<< 129264002 | Action (qualifier value) |)]], [[0..*]] 260870009 | Priority | = [[+scg(<< 272125009 | Priorities (qualifier value) |)]], [[0..*]] 405815000 | Procedure device | = [[+scg(<< 49062001 | Device (physical object) |)]], [[0..*]] 405816004 | Procedure morphology | = [[+scg(<< 49755003 | Morphologically abnormal structure (morphologic abnormality) |)]], [[0..*]] 363704007 | Procedure site | = [[+scg(<< 442083009 | Anatomical or acquired body structure (body structure) |)]], [[0..*]] 405813007 | Procedure site - Direct | = [[+scg(<< 442083009 | Anatomical or acquired body structure (body structure) |)]], [[0..*]] 405814001 | Procedure site - Indirect | = [[+scg(<< 442083009 | Anatomical or acquired body structure (body structure) |)]], [[0..*]] 370131001 | Recipient category | = [[+scg(<< 125676002 | Person (person) | OR << 35359004 | Family (social concept) | OR << 133928008 |	http://snomed.org/dom71388002

refsetId	referenced Component Id	domain Constraint	parent Domain	proximal Primitive Constraint	proximal Primitive Refinement	domainTemplateForPrecoordination	domainTemplateForPostcoordination	guideURL
						Community (social concept) | OR << 105455006 | Donor for medical or surgical procedure (person) | OR << 389109008 | Group (social concept) |)]], [[0..*]] 246513007 | Revision status | = [[+id(<< 261424001 | Primary operation (qualifier value) | OR << 255231005 | Revision - value (qualifier value) | OR << 257958009 | Part of multistage procedure (qualifier value) |)]], [[0..*]] 425391005 | Using access device | = [[+id(<< 49062001 | Device (physical object) |)]], [[0..*]] 424226004 | Using device | = [[+id(<< 49062001 | Device (physical object) |)]], [[0..*]] 424244007 | Using energy | = [[+id(<< 78621006 | Physical force (physical force) |)]], [[0..*]] 424361007 | Using substance | = [[+id(<< 105590001 | Substance (substance) |)]] }	Community (social concept) | OR << 105455006 | Donor for medical or surgical procedure (person) | OR << 389109008 | Group (social concept) |)]], [[0..*]] 246513007 | Revision status | = [[+scg(<< 261424001 | Primary operation (qualifier value) | OR << 255231005 | Revision - value (qualifier value) | OR << 257958009 | Part of multistage procedure (qualifier value) |)]], [[0..*]] 425391005 | Using access device | = [[+scg(<< 49062001 | Device (physical object) |)]], [[0..*]] 424226004 | Using device | = [[+scg(<< 49062001 | Device (physical object) |)]], [[0..*]] 424244007 | Using energy | = [[+scg(<< 78621006 | Physical force (physical force) |)]], [[0..*]] 424361007 | Using substance | = [[+scg(<< 105590001 | Substance (substance) |)]] }	
723560006 | MRCM domain international reference set |	386053000 | Evaluation procedure (procedure) |	<< 386053000 | Evaluation procedure (procedure) |	71388002 | Procedure (procedure) |	<< 71388002 | Procedure (procedure) |	[[1..*]] 260686004 | Method | = [[+ (<< 129265001 | Evaluation - action |)]]	[[+id(<< 71388002 | Procedure (procedure) |)]]: [[0..*]] { [[1..*]] 260686004 | Method | = [[+id(<< 129265001 | Evaluation - action |)]], [[0..*]] 246093002 | Component | = [[+id(<< 123037004 | Body structure | OR << 410607006 | Organism | OR << 105590001 | Substance | OR << 123038009 | Specimen | OR << 260787004 | Physical object | OR << 373873005 | Pharmaceutical / biologic product | OR << 419891008 | Record artifact | OR << 363787002 | Observable entity |)]], [[0..*]] 116686009 | Has specimen | = [[+id(<< 123038009 | Specimen (specimen) |)]], [[0..*]] 370129005 | Measurement method | = [[+id(<< 127789004 | Laboratory procedure categorized by method (procedure) |)]], [[0..*]] 370130000 | Property | = [[+id(<< 118598001 | Property of measurement (qualifier value) |)]], [[0..*]] 370132008 | Scale type | = [[+id(<< 30766002 | Quantitative | OR << 26716007 | Qualitative | OR << 117363000 | Ordinal value | OR << 117365007 | Ordinal or quantitative value | OR << 117362005 | Nominal value | OR << 117364006 | Narrative value | OR << 117444000 | Text value |)]], [[0..*]] 370134009 | Time aspect | = [[+id(<<	[[+scg(<< 71388002 | Procedure (procedure) |)]]: [[0..*]] { [[1..*]] 260686004 | Method | = [[+scg(<< 129265001 | Evaluation - action |)]], [[0..*]] 246093002 | Component | = [[+scg(<< 123037004 | Body structure | OR << 410607006 | Organism | OR << 105590001 | Substance | OR << 123038009 | Specimen | OR << 260787004 | Physical object | OR << 373873005 | Pharmaceutical / biologic product | OR << 419891008 | Record artifact | OR << 363787002 | Observable entity |)]], [[0..*]] 116686009 | Has specimen | = [[+scg(<< 123038009 | Specimen (specimen) |)]], [[0..*]] 370129005 | Measurement method | = [[+scg(<< 127789004 | Laboratory procedure categorized by method (procedure) |)]], [[0..*]] 370130000 | Property | = [[+scg(<< 118598001 | Property of measurement (qualifier value) |)]], [[0..*]] 370132008 | Scale type | = [[+scg(<< 30766002 | Quantitative | OR << 26716007 | Qualitative | OR << 117363000 | Ordinal value | OR << 117365007 | Ordinal or quantitative value | OR << 117362005 | Nominal value | OR << 117364006 | Narrative value | OR << 117444000 | Text value |)]], [[0..*]] 370134009 | Time aspect | = [[+scg(<<	http://snomed.org/dom386053000

refsetId	referenced Component Id	domain Constraint	parent Domain	proximal Primitive Constraint	proximal Primitive Refinement	domainTemplateForPrecoordination	domainTemplateForPostcoordination	guideURL
						7389001 | Time frame (qualifier value) |)]], [[0..*]] 260507000 | Access | = [[+id(<< 309795001 | Surgical access values (qualifier value) |)]], [[0..*]] 363699004 | Direct device | = [[+id(<< 49062001 | Device (physical object) |)]], [[0..*]] 363700003 | Direct morphology | = [[+id(<< 49755003 | Morphologically abnormal structure (morphologic abnormality) |)]], [[0..*]] 363701004 | Direct substance | = [[+id(<< 105590001 | Substance (substance) | OR << 373873005 | Pharmaceutical / biologic product (product) |)]], [[0..*]] 363702006 | Has focus | = [[+id(<< 404684003 | Clinical finding (finding) | OR << 71388002 | Procedure (procedure) |)]], [[0..*]] 363703001 | Has intent | = [[+id(<< 363675004 | Intents (nature of procedure values) (qualifier value) |)]], [[0..*]] 363710007 | Indirect device | = [[+id(<< 49062001 | Device (physical object) |)]], [[0..*]] 363709002 | Indirect morphology | = [[+id(<< 49755003 | Morphologically abnormal structure (morphologic abnormality) |)]], [[0..*]] 260686004 | Method | = [[+id(<< 129264002 | Action (qualifier value) |)]], [[0..*]] 260870009 | Priority | = [[+id(<< 272125009 | Priorities (qualifier value) |)]], [[0..*]] 405815000 | Procedure device | = [[+id(<< 49062001 | Device (physical object) |)]], [[0..*]] 405816004 | Procedure morphology | = [[+id(<< 49755003 | Morphologically abnormal structure (morphologic abnormality) |)]], [[0..*]] 363704007 | Procedure site | = [[+id(<< 442083009 | Anatomical or acquired body structure (body structure) |)]], [[0..*]] 405813007 | Procedure site - Direct | = [[+id(<< 442083009 | Anatomical or acquired body structure (body structure) |)]], [[0..*]] 405814001 | Procedure site - Indirect | = [[+id(<< 442083009 | Anatomical or acquired body structure (body structure) |)]], [[0..*]] 370131001 | Recipient category | = [[+id(<< 125676002 | Person (person) | OR << 35359004 | Family (social concept) | OR << 133928008 | Community (social concept) | OR << 105455006 | Donor for medical or surgical procedure (person) | OR << 389109008 | Group (social concept) |)]], [[0..*]] 246513007 | Revision status | = [[+id(<< 261424001 | Primary operation (qualifier value) | OR << 255231005 | Revision - value (qualifier value) | OR << 257958009 | Part of multistage procedure	7389001 | Time frame (qualifier value) |)]], [[0..*]] 260507000 | Access | = [[+scg(<< 309795001 | Surgical access values (qualifier value) |)]], [[0..*]] 363699004 | Direct device | = [[+scg(<< 49062001 | Device (physical object) |)]], [[0..*]] 363700003 | Direct morphology | = [[+scg(<< 49755003 | Morphologically abnormal structure (morphologic abnormality) |)]], [[0..*]] 363701004 | Direct substance | = [[+scg(<< 105590001 | Substance (substance) | OR << 373873005 | Pharmaceutical / biologic product (product) |)]], [[0..*]] 363702006 | Has focus | = [[+scg(<< 404684003 | Clinical finding (finding) | OR << 71388002 | Procedure (procedure) |)]], [[0..*]] 363703001 | Has intent | = [[+scg(<< 363675004 | Intents (nature of procedure values) (qualifier value) |)]], [[0..*]] 363710007 | Indirect device | = [[+scg(<< 49062001 | Device (physical object) |)]], [[0..*]] 363709002 | Indirect morphology | = [[+scg(<< 49755003 | Morphologically abnormal structure (morphologic abnormality) |)]], [[0..*]] 260686004 | Method | = [[+scg(<< 129264002 | Action (qualifier value) |)]], [[0..*]] 260870009 | Priority | = [[+scg(<< 272125009 | Priorities (qualifier value) |)]], [[0..*]] 405815000 | Procedure device | = [[+scg(<< 49062001 | Device (physical object) |)]], [[0..*]] 405816004 | Procedure morphology | = [[+scg(<< 49755003 | Morphologically abnormal structure (morphologic abnormality) |)]], [[0..*]] 363704007 | Procedure site | = [[+scg(<< 442083009 | Anatomical or acquired body structure (body structure) |)]], [[0..*]] 405813007 | Procedure site - Direct | = [[+scg(<< 442083009 | Anatomical or acquired body structure (body structure) |)]], [[0..*]] 405814001 | Procedure site - Indirect | = [[+scg(<< 442083009 | Anatomical or acquired body structure (body structure) |)]], [[0..*]] 370131001 | Recipient category | = [[+scg(<< 125676002 | Person (person) | OR << 35359004 | Family (social concept) | OR << 133928008 | Community (social concept) | OR << 105455006 | Donor for medical or surgical procedure (person) | OR << 389109008 | Group (social concept) |)]], [[0..*]] 246513007 | Revision status | = [[+scg(<< 261424001 | Primary operation (qualifier value) | OR << 255231005 | Revision - value (qualifier value) | OR << 257958009 | Part of	

5.2.17 MRCM Attribute Domain Reference Set

Purpose

An 723604009 |MRCM attribute domain reference set| allows attributes to be associated with the domains in which they may be applied. It also allows grouping and cardinality constraints to be specified for each attribute and domain combination. For each attribute-domain rule, the strength of the rule (e.g. 723597001 |Mandatory concept model rule| or 723598006 |Optional concept model rule|) and the content type over which this rule applies (e.g. 723596005 |All SNOMED CT content|, 723594008 |All precoordinated SNOMED CT content|) is also specified.

Each attribute is identified by its concept id, while each domain is identified by the same concept id used in the referencedComponentId of the 723589008 |MRCM domain reference set|.

Data Structure

An 723604009 |MRCM attribute domain reference set| is structured as shown in the following table.

Field	**Data type**	**Purpose**	Mutable	**Part of Primary Key**
id	UUID	A 128 bit unsigned Integer, uniquely identifying this reference set member. Different versions of a *reference set member* share the same id but have different effectiveTime. This allows a *reference set member* to be modified or made inactive (i.e. removed from the active set) at a specified time.	NO	YES (Full / Snapshot)
effectiveTime	Time	The inclusive date or time at which this version of the identified reference set member became the current version. **Note** : In distribution files the effectiveTime should follow the short ISO date format (*YYYYMMDD*) and should not include the hours, minutes, seconds or timezone indicator. The current version of this reference set member at time *T* is the version with the most recent effectiveTime prior to or equal to time *T* .	YES	YES (Full) Optional (Snapshot)
active	Boolean	The state of the identified reference set member as at the specified effectiveTime . If active = 1 (true) the reference set member is part of the current version of the set, if active = 0 (false) the reference set member is not part of the current version of the set.	YES	NO
moduleId	SCTID	Identifies the SNOMED CT module that contains this reference set member as at the specified effectiveTime . The value must be a subtype of 900000000000443000 | Module (core metadata concept) | within the metadata hierarchy.	YES	NO
refsetId	SCTID	Identifies the reference set to which this reference set member belongs. In this case, a subtype descendant of: 723604009 | MRCM attribute domain reference set |	NO	NO
referencedComponentId	SCTID	A reference to the SNOMED CT component to be included in the reference set. A reference to the SNOMED CT attribute concept to which the attribute-domain rule defined by this member applies.	NO	NO

domainId	SCTID	A reference to the SNOMED CT concept that identifies the relevant concept domain.	NO	NO
grouped	Boolean	Whether or not the given attribute (identified by referencedComponentId) is treated by a Description Logic reasoner as belonging to a relationship group, when applied to a concept in the given domain. If grouped = 1 (true) then the given attribute (identified by referencedComponentId) is treated by a Description Logic reasoner as belonging to a relationship group. If grouped = 0 (false) then the given attribute (identified by referencedComponentId) is treated by a Description Logic reasoner as not belonging to a relationship group.	YES	NO
attributeCardinality	string	The number of times the given attribute can be assigned a distinct (non-redundant) value within the definition of each concept or expression. This string can be parsed using the following ABNF rule (together with the subrules defined in the Expression Constraint Language): *attributeCardinality = minimum to maximum*	YES	NO
attributeInGroupCardinality	string	The number of times the given attribute can be assigned a distinct (non-redundant) value within a single relationship group as part of the definition of a concept or expression. This string can be parsed using the following ABNF rule (together with the subrules defined in the Expression Constraint Language): *attributeCardinality = minimum to maximum*	YES	NO
ruleStrengthId	SCTID	A subtype of 723573005 |Concept model rule strength| which specifies whether the given rule is mandatory (resulting in an error) or optional (resulting in a warning).	YES	NO
contentTypeId	SCTID	A subtype of 723574004 |Content type| which indicates the type of SNOMED CT content over which this rule applies. In many cases, this will be set to 723596005 |All SNOMED CT content|.	YES	NO

Metadata

The following metadata hierarchy supports this reference set:

- 900000000000454005 |Foundation metadata concept|
    - 900000000000455006 |Reference set|
        - 723564002 |MRCM reference set|
            - 723604009 |MRCM attribute domain reference set|
                - 723561005 |MRCM attribute domain international reference set|
    - 900000000000457003 |Reference set attribute|
        - 723571007 |Cardinality|
            - 723602008 |Attribute cardinality|
            - 723603003 |Attribute in group cardinality|
        - 723574004 |Content type| [1]
            - 723593002 |All new precoordinated SNOMED CT content|
                - 723594008 |All precoordinated SNOMED CT content|
                    - 723596005 |All SNOMED CT content|
            - 723595009 |All postcoordinated SNOMED CT content|
                - 723596005 |All SNOMED CT content|
        - 609431004 |Domain|
        - 723572000 |Grouped|
        - 723573005 |Concept model rule strength|

- 723597001 |Mandatory concept model rule|
- 723598006 |Optional concept model rule|

Please note that the 723574004 |Content type| hierarchy is designed using 'universal restriction' logic. The hierarchy may therefore appear to be 'upside down'. However, it was designed in this way because if an MRCM rule applies to 723596005 |All SNOMED CT content| then it also applies to the Content Types that are a supertype of this - including 723594008 |All precoordinated SNOMED CT content| and 723595009 |All postcoordinated SNOMED CT content| .

Reference Set Descriptor and Example Data

Notes on the tables used to show descriptors and examples
The reference set example tables on this page have been revised as follows to aid clarity and understanding:

- The first four columns which are present in all release files are not shown. The omitted columns (id, effectiveTime, active, moduleId) are used in the same way in all referenced sets to support identification, versioning and packaging. They do not directly affect the specific features of a particular reference set or reference set type.
- Reference set columns that contain SNOMED CT identifiers are expanded to show details of the concept or description referenced by that identifier. In some cases, the term is shown in the same column using the expression syntax, in other cases an additional column with a name suffix '_term' has been added. In the standard reference set files only the identifier is present in the column and there is no added column for the term. When using reference sets, the term and other details of the component are looked up from the relevant component release files.

Descriptor Template

The table below shows the reference set descriptor for a reference set that follows the 723604009 |MRCM attribute domain reference set| pattern.

refsetId	referencedComponentId	attributeDescription	attributeType	attribute Order
900000000000456007 |Reference set descriptor|	723604009 |MRCM attribute domain reference set|	449608002 |Referenced component|	900000000000461009 |Concept type component|	0
900000000000456007 |Reference set descriptor|	723604009 |MRCM attribute domain reference set|	609431004 |Domain|	900000000000461009 |Concept type component|	1
900000000000456007 |Reference set descriptor|	723604009 |MRCM attribute domain reference set|	723572000 |Grouped|	900000000000478000 |Unsigned integer|	2
900000000000456007 |Reference set descriptor|	723604009 |MRCM attribute domain reference set|	723602008 |Attribute Cardinality|	707000009 |SNOMED CT parsable string|	3
900000000000456007 |Reference set descriptor|	723604009 |MRCM attribute domain reference set|	723603003 |Attribute In Group Cardinality|	707000009 |SNOMED CT parsable string|	4
900000000000456007 |Reference set descriptor|	723604009 |MRCM attribute domain reference set|	723573005 |Concept model rule Strength|	900000000000461009 |Concept type component|	5

refsetId	referencedComponentId	attributeDescription	attributeType	attribute Order
900000000000456007 | Reference set descriptor |	723604009 | MRCM attribute domain reference set |	723574004 | Content type |	900000000000461009 | Concept type component |	6

Example Data

The table below shows some example rows from a reference set that follows the format of the 723604009 |MRCM attribute domain reference set| .

refsetId	referencedComponent Id	domainId	grouped	attribute Cardinality	attribute InGroup Cardinality	ruleStrengthId	contentTypeId
723561005 | MRCM attribute domain international reference set |	255234002 | After |	404684003 | Clinical finding (finding) |	1	0..*	0..*	723597001 | Mandatory concept model rule |	723596005 | All SNOMED CT content |
723561005 | MRCM attribute domain international reference set |	255234002 | After |	272379006 | Event (event) |	1	0..*	0..*	723597001 | Mandatory concept model rule |	723596005 | All SNOMED CT content |
723561005 | MRCM attribute domain international reference set |	408729009 | Finding context |	413350009 | Finding with explicit context (situation) |	1	0..*	0..1	723597001 | Mandatory concept model rule |	723596005 | All SNOMED CT content |
723561005 | MRCM attribute domain international reference set |	272741003 | Laterality |	91723000 | Anatomical structure (body structure) |	0	0..1	0..0	723597001 | Mandatory concept model rule |	723594008 | All precoordinated SNOMED CT content |

5.2.18 MRCM Attribute Range Reference Set

Purpose

An 723592007 |MRCM attribute range reference set| allows attributes to be associated with a valid value range for a given SNOMED CT content type and rule strength. The range of each attribute is defined using an Expression Constraint. This expression constraint represents the set of concepts, expressions, or concrete values that may be used as the value of the given attribute.[1]
The 723592007 |MRCM attribute range reference set|also provides a summary of the concept model rule associated with each attribute (including all valid domains and the given range) using an Expression Constraint representation. This attribute rule can be completely auto-generated by combining information from the 723604009 |MRCM attribute domain reference set| and the 723592007 |MRCM attribute range reference set| .

Data Structure

An 723592007 |MRCM attribute range reference set| is structured as shown in the following table.

Field	**Data type**	**Purpose**	Mutable	**Part of Primary Key**
id	UUID	A 128 bit unsigned Integer, uniquely identifying this reference set member. Different versions of a *reference set member* share the same id but have different effectiveTime. This allows a *reference set member* to be modified or made inactive (i.e. removed from the active set) at a specified time.	NO	YES (Full / Snapshot)
effectiveTime	Time	The inclusive date or time at which this version of the identified reference set member became the current version. **Note** : In distribution files the effectiveTime should follow the short ISO date format (*YYYYMMDD*) and should not include the hours, minutes, seconds or timezone indicator. The current version of this reference set member at time *T* is the version with the most recent effectiveTime prior to or equal to time *T* .	YES	YES (Full) Optional (Snapshot)
active	Boolean	The state of the identified reference set member as at the specified effectiveTime . If active = 1 (true) the reference set member is part of the current version of the set, if active = 0 (false) the reference set member is not part of the current version of the set.	YES	NO
moduleId	SCTID	Identifies the SNOMED CT module that contains this reference set member as at the specified effectiveTime . The value must be a subtype of 900000000000443000 | Module (core metadata concept) | within the metadata hierarchy.	YES	NO
refsetId	SCTID	Identifies the reference set to which this reference set member belongs. In this case, a subtype descendant of: 723592007 | MRCM attribute range reference set |	NO	NO
referencedComponentId	SCTID	A reference to the SNOMED CT component to be included in the reference set. A reference to the SNOMED CT attribute concept to which the range defined by this member applies.	NO	NO

rangeConstraint	string	An expression constraint, which defines the set of concepts that may be used as the value of the given attribute (identified by referencedComponentId). This string can be parsed using the ABNF syntax defined for the Expression Constraint Language . If ranges using concrete values are required, the Expression Constraint Language can be extended with the keyword "TYPE", by replacing the **simpleExpressionConstraint** rule with the following two rules: **simpleExpressionConstraint** = [constraintOperator ws] eclFocusConcept / typeKeyword ws conceptReference **typeKeyword** = ("t"/"T") ("y"/"Y") ("p"/"P") ("e"/"E") For example, the following range includes the set of all integers: TYPE 900000000000476001 |Integer| Any descendant of 900000000000459000 |Attribute type| may be used as the type of an attribute range.	YES	NO
attributeRule	string	An Expression Constraint that captures the domain, range and cardinality constraints for the given attribute, rule strength and content type. This string can be parsed using the ABNF syntax defined for the Expression Constraint Language . If ranges with concrete values are required, the Expression Constraint Language can be extended as described above (for rangeConstraint).	YES	NO
ruleStrengthId	SCTID	A subtype of 723573005 |Concept model rule strength| which specifies whether the given rule is mandatory (resulting in an error) or optional (resulting in a warning).	YES	NO
contentTypeId	SCTID	A subtype of 723574004 |Content type| which indicates the type of SNOMED CT content over which this rule applies.	YES	NO

Metadata

The following metadata hierarchy supports this reference set:

- 900000000000454005 |Foundation metadata concept|
    - 900000000000455006 |Reference set|
        - 723564002 |MRCM reference set|
            - 723592007 |MRCM attribute range reference set|
                - 723562003 |MRCM attribute range international reference set|
    - 900000000000457003 |Reference set attribute|
        - 723576002 |Attribute rule|
        - 723574004 |Content type|[2]
            - 723593002 |All new precoordinated SNOMED CT content|
                - 723594008 |All precoordinated SNOMED CT content|
                    - 723596005 |All SNOMED CT content|
            - 723595009 |All postcoordinated SNOMED CT content|
                - 723596005 |All SNOMED CT content|
        - 723575003 |Range constraint|
        - 723573005 |Concept model rule strength|
            - 723597001 |Mandatory concept model rule|
            - 723598006 |Optional concept model rule|

[1] If ranges including concrete values (such as integers or strings) are required, the Expression Constraint Language can be extended, as described for *rangeConstraint* in the Data Structure section on this page.

Please note that the 723574004 |Content type| hierarchy is designed using 'universal restriction' logic. The hierarchy may therefore appear to be 'upside down'. However, it was designed in this way because if an MRCM rule applies to 723596005 |All SNOMED CT content| then it also applies to the Content Types that are a supertype of this - including 723594008 |All precoordinated SNOMED CT content| and 723595009 |All postcoordinated SNOMED CT content| .

Reference Set Descriptor and Example Data

Notes on the tables used to show descriptors and examples
The reference set example tables on this page have been revised as follows to aid clarity and understanding:

- The first four columns which are present in all release files are not shown. The omitted columns (id, effectiveTime, active, moduleId) are used in the same way in all referenced sets to support identification, versioning and packaging. They do not directly affect the specific features of a particular reference set or reference set type.
- Reference set columns that contain SNOMED CT identifiers are expanded to show details of the concept or description referenced by that identifier. In some cases, the term is shown in the same column using the expression syntax, in other cases an additional column with a name suffix '_term' has been added. In the standard reference set files only the identifier is present in the column and there is no added column for the term. When using reference sets, the term and other details of the component are looked up from the relevant component release files.

Descriptor Template

The table below shows the reference set descriptor for a reference set that follows the 723592007 |MRCM attribute range reference set| pattern.

refsetId	referencedComponentId	attributeDescription	attributeType	attribute Order
900000000000456007 |Reference set descriptor|	723592007 |MRCM attribute range reference set|	449608002 |Referenced component|	900000000000461009 |Concept type component|	0
900000000000456007 |Reference set descriptor|	723592007 |MRCM attribute range reference set|	723575003 |Range constraint|	707000009 |SNOMED CT parsable string|	1
900000000000456007 |Reference set descriptor|	723592007 |MRCM attribute range reference set|	723576002 |Attribute rule|	707000009 |SNOMED CT parsable string|	2
900000000000456007 |Reference set descriptor|	723592007 |MRCM attribute range reference set|	723573005 |Concept model rule strength|	900000000000461009 |Concept type component|	3
900000000000456007 |Reference set descriptor|	723592007 |MRCM attribute range reference set|	723574004 |Content type|	900000000000461009 |Concept type component|	4

Example Data

The table below shows some example rows from a reference set that follows the format of the 723592007 |MRCM attribute range reference set|.

refsetId	referencedComponentId	rangeConstraint	attributeRule	ruleStrengthId	contentTypeId
723562003 | MRCM attribute range international reference set |	255234002 | After |	<< 404684003 | Clinical finding (finding) | OR << 71388002 | Procedure (procedure) |	(<< 404684003 | Clinical finding (finding) | OR << 272379006 | Event (event) |): [0..*] { [0..*] 255234002 | After | = (<< 404684003 | Clinical finding (finding) | OR << 71388002 | Procedure (procedure) |)}	723597001 | Mandatory concept model rule |	723596005 | All SNOMED CT content |
723562003 | MRCM attribute range international reference set |	408729009 | Finding context |	<< 410514004 | Finding context value (qualifier value) |	<< 413350009 | Finding with explicit context (situation) | : [0..*] { [0..1] 408729009 | Finding context | = << 410514004 | Finding context value (qualifier value) | }	723597001 | Mandatory concept model rule |	723596005 | All SNOMED CT content |
723562003 | MRCM attribute range international reference set |	272741003 | Laterality |	<< 182353008 | Side (qualifier value) |	<< 91723000 | Anatomical structure (body structure) | : [0..1] 272741003 | Laterality | = << 182353008 | Side (qualifier value) |	723597001 | Mandatory concept model rule |	723596005 | All SNOMED CT content |

5.2.19 MRCM Module Scope Reference Set

Purpose

An 723563008 |MRCM module scope reference set| specifies the set of MRCM reference sets that should be applied to the content in each module. Within a SNOMED CT Edition, the MRCM rules applied to the included modules must be consistent, to ensure data integrity within an edition is maintained.

Data Structure

An 723563008 |MRCM module scope reference set| is structured as shown in the following table.

Field	**Data type**	**Purpose**	Mutable	**Part of Primary Key**
id	UUID	A 128 bit unsigned Integer, uniquely identifying this reference set member. Different versions of a *reference set member* share the same id but have different effectiveTime. This allows a *reference set member* to be modified or made inactive (i.e. removed from the active set) at a specified time.	NO	YES (Full / Snapshot)
effectiveTime	Time	The inclusive date or time at which this version of the identified reference set member became the current version. **Note** : In distribution files the effectiveTime should follow the short ISO date format (*YYYYMMDD*) and should not include the hours, minutes, seconds or timezone indicator. The current version of this reference set member at time *T* is the version with the most recent effectiveTime prior to or equal to time *T* .	YES	YES (Full) Optional (Snapshot)
active	Boolean	The state of the identified reference set member as at the specified effectiveTime . If active = 1 (true) the reference set member is part of the current version of the set, if active = 0 (false) the reference set member is not part of the current version of the set.	YES	NO
moduleId	SCTID	Identifies the SNOMED CT module that contains this reference set member as at the specified effectiveTime . The value must be a subtype of 900000000000443000 | Module (core metadata concept) | within the metadata hierarchy.	YES	NO
refsetId	SCTID	Identifies the reference set to which this reference set member belongs. In this case, set to 723563008 | MRCM module scope reference set |	NO	NO
referencedComponentId	SCTID	A reference to the SNOMED CT component to be included in the reference set. Identifies the SNOMED CT module to which the given concept model refset is applied. The value must be a subtype of 900000000000443000 | Module | within the metadata hierarchy.	NO	NO
mrcmRuleRefsetId	SCTID	A subtype of 723564002 | MRCM reference set | that defines the concept model rules that are applied to content in the module identified by referencedComponentId.	NO	NO

Metadata

The following metadata hierarchy supports this reference set:

- 900000000000454005 |Foundation metadata concept|
    - 900000000000455006 |Reference set|
        - 723564002 |MRCM reference set|
            - 723563008 |MRCM module scope reference set|
    - 900000000000457003 |Reference set attribute|
        - 723577006 |MRCM rule reference set|

Reference Set Descriptor and Example Data

Notes on the tables used to show descriptors and examples
The reference set example tables on this page have been revised as follows to aid clarity and understanding:

- The first four columns which are present in all release files are not shown. The omitted columns (id, effectiveTime, active, moduleId) are used in the same way in all referenced sets to support identification, versioning and packaging. They do not directly affect the specific features of a particular reference set or reference set type.
- Reference set columns that contain SNOMED CT identifiers are expanded to show details of the concept or description referenced by that identifier. In some cases, the term is shown in the same column using the expression syntax, in other cases an additional column with a name suffix '_term' has been added. In the standard reference set files only the identifier is present in the column and there is no added column for the term. When using reference sets, the term and other details of the component are looked up from the relevant component release files.

Descriptor Template

The table below shows the reference set descriptor for the 723563008 |MRCM module scope reference set| pattern.

refsetId	referencedComponentId	attributeDescription	attributeType	attributeOrder
900000000000456007 |Reference set descriptor|	723563008 |MRCM module scope reference set|	449608002 |Referenced component|	900000000000461009 |Concept type component|	0
900000000000456007 |Reference set descriptor|	723563008 |MRCM module scope reference set|	723577006 |MRCM rule reference set|	900000000000461009 |Concept type component|	1

Example Data

The table below shows some example rows from the 723563008 |MRCM module scope reference set|.

refsetId	referencedComponentId	mrcmRuleRefsetId
723563008 |MRCM module scope reference set|	900000000000207008 |SNOMED CT core module (core metadata concept)|	723560006 |MRCM domain international reference set|
723563008 |MRCM module scope reference set|	900000000000207008 |SNOMED CT core module (core metadata concept)|	723561005 |MRCM attribute domain international reference set|
723563008 |MRCM module scope reference set|	900000000000207008 |SNOMED CT core module (core metadata concept)|	723562003 |MRCM attribute range international reference set|

5.2.20 Ordered Reference Set

Deprecation Notice
The Ordered Reference Set pattern is now deprecated as it has been replaced with two reference set types each of which is specific to one of the two distinct use cases supported by the Ordered Reference set pattern.
The recommended Reference sets to address the purposes identified below are now:

- 5.2.2 Ordered Component Reference Set
  - This allows an ordered or prioritized list of components to be represented.
  - It omits the **linkedToId** field in the pattern shown below as this is not required to address this use case.
- 5.2.6 Ordered Association Reference Set
  - This enables representation of alternative navigation hierarchies (in which child concepts are ordered) and also also supports representation of groups of ordered components.
  - The **linkedToId** field in the pattern shown below is replaced by the targetComponentId (this name is used to align with the 5.2.5 Association Reference Set (used from unordered associations).

Deprecation does not prevent continued use of an existing reference set pattern. However it does indicate that a different solution is now specified and recommended to meet the requirements for this pattern

Purpose

An 447258008 |Ordered type reference set| allows a collection of components to be defined with a specified given a priority ordering. This type of reference ret can also be used to specify ordered associations between different components. These can be used to specify several interrelated subsets of components and to define alternative hierarchies for navigation and selection of concepts or descriptions.

Data structure

An Ordered reference set is an Integer Component reference set is used to represent ordered lists and alternative hierarchies. Its structure is shown in the following table.

Table 5.2.20-4: Ordered reference set - Data structure

Field	Data type	Purpose
id	UUID	A 128 bit unsigned Integer, uniquely identifying this reference set member. Different versions of a *reference set member* share the same id but have different effectiveTime. This allows a *reference set member* to be modified or made active (i.e. removed from the active set) at a specified time.
effectiveTime	Time	The inclusive date or time at which this version of the identified reference set member became the current version. The current version of this reference set member at time *T* is the version with the most recent effectiveTime prior to or equal to time *T* .
active	Boolean	The state of the identified reference set member as at the specified effectiveTime . If active = 1 (true) the reference set member is part of the current version of the set, if active = 0 (false) the reference set member is not part of the current version of the set.

Field	Data type	Purpose
moduleId	SCTID	Identifies the SNOMED CT module that contains this reference set member as at the specified effectiveTime . The value must be a subtype of 900000000000443000 |Module (core metadata concept)| within the metadata hierarchy.
refsetId	SCTID	Identifies the reference set to which this reference set member belongs. In this case, set to a subtype of 447258008 |Ordered type reference set|
referencedComponentId	SCTID	The identifier of a SNOMED CT component that is included in the ordered list of alternative hierarchy.
order	Integer	Specifies the sort order of the list. The list is ordered by applying an ascending sort of the order value. The value of order =1 represents the highest priority. A value of '0' is not allowed. Duplicate values are permitted and the sort order between two members with the same order value is not defined. If the linkedToId value is not 0, sorting occurs within subgroups that share the same linkedToId. Note: The name "order" is a reserved word in some database environments. Please consider this when using this column.
linkedToId	SCTID	The identifier of a SNOMED CT component that acts as a grouper or hierarchy node, collecting together a subgroup from within the list. This field either enables reference set member linked into a number of subgroups. These subgroups can be nested allowing representation of alternative hierarchies. To link members into a subgroup, all components in the same subgroup should reference the same component. This can either be a component that represents the name of that subgroup or the first member of the subgroup. In the latter case, the first row of each subgroup will contain the same identifier in referencedComponentId and linkedToId and with order =1. To link a number of children concepts to a single parent concept, one member record should exist per child, with the referencedComponentId field referencing the parent and this field referencing the child concept. The order field is then used to order the children concepts under the parent concept. For ordered lists that do not require grouping or hierarchical arrangement the value of linkedToId should be the digit zero (0).

Metadata

The following metadata in the "Foundation metadata concept" hierarchy supports this reference set

Table 5.2.20-4: Ordered References Sets in the Metadata Hierarchy

Field	Data type	Purpose	Mutable	Part of Primary Key

id	UUID	A 128 bit unsigned Integer, uniquely identifying this reference set member. Different versions of a *reference set member* share the same id but have different effectiveTime. This allows a *reference set member* to be modified or made inactive (i.e. removed from the active set) at a specified time.	NO	YES (Full / Snapshot)
effectiveTime	Time	The inclusive date or time at which this version of the identified reference set member became the current version. **Note** : In distribution files the effectiveTime should follow the short ISO date format (*YYYYMMDD*) and should not include the hours, minutes, seconds or timezone indicator. The current version of this reference set member at time *T* is the version with the most recent effectiveTime prior to or equal to time *T* .	YES	YES (Full) Optional (Snapshot)
active	Boolean	The state of the identified reference set member as at the specified effectiveTime . If active = 1 (true) the reference set member is part of the current version of the set, if active = 0 (false) the reference set member is not part of the current version of the set.	YES	NO
moduleId	SCTID	Identifies the SNOMED CT module that contains this reference set member as at the specified effectiveTime . The value must be a subtype of 900000000000443000 | Module (core metadata concept) | within the metadata hierarchy.	YES	NO
refsetId	SCTID	Identifies the reference set to which this reference set member belongs. In this case, a subtype descendant of: 447258008 | Ordered type reference set |	NO	NO
referencedComponentId	SCTID	A reference to the SNOMED CT component to be included in the reference set. The component that is placed in order by this reference set row.	NO	NO
order	Integer	Specifies the sort order of the list. The list is ordered by applying an ascending sort of the order value. The value of order =1 represents the highest priority. A value of '0' is not allowed. Duplicate values are permitted and the sort order between two members with the same order value is not defined. If the linkedToId value is not 0, sorting occurs within subgroups that share the same linkedToId. Note: The name "order" is a reserved word in some database environments. Please consider this when using this column.	YES	NO

linkedToId	SCTID	The identifier of a SNOMED CT component that acts as a grouper or hierarchy node, collecting together a subgroup from within the list. This field either enables reference set member linked into a number of subgroups. These subgroups can be nested allowing representation of alternative hierarchies. To link members into a subgroup, all components in the same subgroup should reference the same component. This can either be a component that represents the name of that subgroup or the first member of the subgroup. In the latter case, the first row of each subgroup will contain the same identifier in referencedComponentId and linkedToId and with order =1. To link a number of children concepts to a single parent concept, one member record should exist per child, with the referencedComponentId field referencing the parent and this field referencing the child concept. The order field is then used to order the children concepts under the parent concept. For ordered lists that do not require grouping or hierarchical arrangement the value of linkedToId should be the digit zero (0).	YES	NO

- 900000000000454005 |Foundation metadata concept|
  - 900000000000455006 |Reference set|
    - 447258008 |Ordered type reference set|

Reference Set Descriptor and Example Data

Notes on the tables used to show descriptors and examples
The reference set example tables on this page have been revised as follows to aid clarity and understanding:

- The first four columns which are present in all release files are not shown. The omitted columns (id, effectiveTime, active, moduleId) are used in the same way in all referenced sets to support identification, versioning and packaging. They do not directly affect the specific features of a particular reference set or reference set type.
- Reference set columns that contain SNOMED CT identifiers are expanded to show details of the concept or description referenced by that identifier. In some cases, the term is shown in the same column using the expression syntax, in other cases an additional column with a name suffix '_term' has been added. In the standard reference set files only the identifier is present in the column and there is no added column for the term. When using reference sets, the term and other details of the component are looked up from the relevant component release files.

Descriptor Template

The tables below show the descriptor that defines the structure of the 447258008 |Ordered type reference set| pattern and an example of descriptor for a reference set that follows this pattern.

Table 5.2.20-4: Refset Descriptor rows for Ordered Reference Set

refsetId	referencedComponentId	attributeDescription	attributeType	attributeOrder
900000000000456007 |Reference set descriptor|	447258008 |Ordered type reference set|	449608002 |Referenced component|	900000000000460005 |Component type|	0
900000000000456007 |Reference set descriptor|	447258008 |Ordered type reference set|	447255006 |Priority order reference set attribute|	900000000000478000 |Unsigned integer|	1

900000000000456007 |Reference set descriptor|	447258008 |Ordered type reference set|	447257003 |"Linked to" reference set attribute|	900000000000460005 |Component type|	2

Example Data

Table 5.2.20-4: Sample Content for an Ordered Reference Set

refsetId	referencedComponentId (Referenced component)	order (Attribute order)	linkedTo ("Linked to" reference set attribute)
447570008 |SNOMED CT top level navigation hierarchy ordered reference set|	64572001 |Disease|	1	123946008 |Disorder by body site|
447570008 |SNOMED CT top level navigation hierarchy ordered reference set|	64572001 |Disease|	2	370117001 |Disorder of system|
447570008 |SNOMED CT top level navigation hierarchy ordered reference set|	64572001 |Disease|	3	278919001 |Communication disorder|
447570008 |SNOMED CT top level navigation hierarchy ordered reference set|	64572001 |Disease|	4	74732009 |Mental disorder|
447570008 |SNOMED CT top level navigation hierarchy ordered reference set|	64572001 |Disease|	5	39898005 |Sleep disorder|
447570008 |SNOMED CT top level navigation hierarchy ordered reference set|	64572001 |Disease|	6	370118006 |Disorder of pregnancy / labor / delivery / puerperium|
447570008 |SNOMED CT top level navigation hierarchy ordered reference set|	64572001 |Disease|	7	370119003 |Fetal / neonatal / perinatal disorder|
447570008 |SNOMED CT top level navigation hierarchy ordered reference set|	64572001 |Disease|	8	370120009 |Endocrine / nutritional / metabolic disorder|
447570008 |SNOMED CT top level navigation hierarchy ordered reference set|	64572001 |Disease|	9	370121008 |Disorder of blood / lymphatics / immune system|
447570008 |SNOMED CT top level navigation hierarchy ordered reference set|	64572001 |Disease|	10	281867008 |Multisystem disorder|

5.2.21 OWL Expression Reference Set

Purpose

An 762676003 |OWL expression type reference set| associates description logic statements with SNOMED CT concept in the OWL functional syntax.

The SNOMED CT International Release contains two reference sets that follow the 762676003 |OWL expression type reference set| pattern:

- The 733073007 |OWL axiom reference set (foundation metadata concept)|, in which the OWL expressions represent and axioms that form part of the definition of the concept identified by the referencedComponentId.
- The 762103008 |OWL ontology reference set (foundation metadata concept)|, in which the OWL expressions represent essential information about an ontology. This information includes, namespaces, ontology URI, ontology version URI, and import statements. The 762103008 |OWL ontology reference set (foundation metadata concept)| enables the use of prefixes in the ontology

Data Structure

An 762676003 |OWL expression type reference set| is structured as shown in the following table.

Field	**Data type**	**Purpose**	Mutable	**Part of Primary Key**
id	UUID	A 128 bit unsigned Integer, uniquely identifying this reference set member. Different versions of a *reference set member* share the same id but have different effectiveTime. This allows a *reference set member* to be modified or made inactive (i.e. removed from the active set) at a specified time.	NO	YES (Full / Snapshot)
effectiveTime	Time	The inclusive date or time at which this version of the identified reference set member became the current version. **Note** : In distribution files the effectiveTime should follow the short ISO date format (*YYYYMMDD*) and should not include the hours, minutes, seconds or timezone indicator. The current version of this reference set member at time *T* is the version with the most recent effectiveTime prior to or equal to time *T* .	YES	YES (Full) Optional (Snapshot)
active	Boolean	The state of the identified reference set member as at the specified effectiveTime . If active = 1 (true) the reference set member is part of the current version of the set, if active = 0 (false) the reference set member is not part of the current version of the set.	YES	NO
moduleId	SCTID	Identifies the SNOMED CT module that contains this reference set member as at the specified effectiveTime . The value must be a subtype of 900000000000443000 | Module (core metadata concept) | within the metadata hierarchy.	YES	NO
refsetId	SCTID	Identifies the reference set to which this reference set member belongs. In this case, a subtype descendant of: 762676003 | OWL expression type reference set (foundation metadata concept) |	NO	NO
referencedComponentId	SCTID	A reference to the SNOMED CT component to be included in the reference set. The concept to which the OWL expression applies. In the case of the 733073007 | OWL axiom reference set (foundation metadata concept) | , the axiom contributes to the definition of the identified concept.	NO	NO
owlExpression	String	The text of OWL expression to attach to the component identified by referencedComponentId .	YES	NO

Metadata

The following metadata supports this reference set:

- 900000000000454005 |Foundation metadata concept|
  - 900000000000455006 |Reference set|
    - 762676003 |OWL expression type reference set|
      - 762103008 |OWL ontology reference set|
      - 733073007 |OWL axiom reference set|
  - 900000000000457003 |Reference set attribute|
    - 706999006 |Expression|
      - 762677007 |OWL expression|
    - 900000000000459000 |Attribute type|
      - 900000000000465000 |String|
      - 762678002 |OWL 2 language syntax|

Descriptor Template and Examples

The reference set example tables on this page have been revised as follows to aid clarity and understanding:

- The first four columns which are present in all release files are not shown. The omitted columns (id, effectiveTime, active) are used in the same way in all referenced sets to support identification, versioning and packaging. They do not directly affect the specific features of a particular reference set or reference set type.
- Reference set columns that contain SNOMED CT identifiers are expanded to show details of the concept or description referenced by that identifier. In some cases, the term is shown in the same column using the expression syntax, in other cases an additional column with a name suffix '_term' has been added. In the standard reference set files only the identifier is present in the column and there is no added column for the term. When using reference sets, the term and other details of the component are looked up from the relevant component release files.

Descriptor Template

The table below shows the descriptors that define the structure of the 762676003 |OWL expression type reference set| pattern and examples of the descriptors for specific reference sets that follow this pattern.

Table 4-3: Descriptor templates for OWL expression reference rets

refsetId	referencedComponentId	attributeDescription	attributeType	attributeOrder
900000000000456007 |Reference set descriptor|	762676003 |OWL expression type reference set|	449608002 |Referenced component|	900000000000461009 |Concept type component|	0
900000000000456007 |Reference set descriptor|	762676003 |OWL expression type reference set|	762677007 |OWL expression|	762678002 |OWL 2 language syntax|	1
900000000000456007 |Reference set descriptor|	762103008 |OWL ontology reference set|	449608002 |Referenced component|	900000000000461009 |Concept type component|	0
900000000000456007 |Reference set descriptor|	762103008 |OWL ontology reference set|	762677007 |OWL expression|	762678002 |OWL 2 language syntax|	1
900000000000456007 |Reference set descriptor|	733073007 |OWL axiom reference set|	449608002 |Referenced component|	900000000000461009 |Concept type component|	0

900000000000456007 |Reference set descriptor|	733073007 |OWL axiom reference set|	762677007 |OWL expression|	762678002 |OWL 2 language syntax|	1

OWL Ontology Reference Set Example

Table 4-3: OWL ontology reference set example

moduleId	**refsetId**	**referencedComponentId**	**owlExpression**
900000000000012004 |SNOMED CT model component module|	762103008 |OWL ontology reference set|	734146004 |OWL ontology namespace|	Prefix(:=<http://snomed.info/id/>)
900000000000012004 |SNOMED CT model component module|	762103008 |OWL ontology reference set|	734146004 |OWL ontology namespace|	Prefix(owl:=<http://www.w3.org/2002/07/owl#>)
900000000000012004 |SNOMED CT model component module|	762103008 |OWL ontology reference set|	734146004 |OWL ontology namespace|	Prefix(rdf:=<http://www.w3.org/1999/02/22-rdf-syntax-ns#>)
900000000000012004 |SNOMED CT model component module|	762103008 |OWL ontology reference set|	734146004 |OWL ontology namespace|	Prefix(xml:=<http://www.w3.org/XML/1998/namespace>)
900000000000012004 |SNOMED CT model component module|	762103008 |OWL ontology reference set|	734146004 |OWL ontology namespace|	Prefix(xsd:=<http://www.w3.org/2001/XMLSchema#>)
900000000000012004 |SNOMED CT model component module|	762103008 |OWL ontology reference set|	734146004 |OWL ontology namespace|	Prefix(rdfs:=<http://www.w3.org/2000/01/rdf-schema#>)
900000000000012004 |SNOMED CT model component module|	762103008 |OWL ontology reference set|	734147008 |OWL ontology header|	Ontology(<http://snomed.info/sct/900000000000207008>)

OWL Axiom Reference Set Example

Table 4-3: OWL axiom reference set example

moduleId	**refsetId**	**referencedComponentId**	**owlExpression**	**Explanatory Notes**
900000000000207008 |SNOMED CT core module|	733073007 |OWL axiom reference set|	404684003 |Clinical finding (finding)|	SubClassOf(:404684003 :138875005)	Example of SubClassOf, which is equivalent to an |Is a| relationship between most SNOMED CT concepts. A different OWL expression is used to represent |Is a| relationships between attributes. This shown in the row below. • 404684003 |Clinical finding (finding)| • 138875005 |SNOMED CT Concept (SNOMED RT+CTV3)|

900000000000012004 |SNOMED CT model component module|	733073007 |OWL axiom reference set|	774081006 |Proper part of (attribute)|	SubObjectPropertyOf(:123005000 :733928003)	Example of SubObjectPropertyOf, which is equivalent to an |Is a| relationship between attributes. • 774081006 |Proper part of (attribute)| • 733928003 |All or part of (attribute)|
900000000000207008 |SNOMED CT core module|	733073007 |OWL axiom reference set|	90708001 |Kidney disease (disorder)|	EquivalentClasses(:90708001 ObjectIntersectionOf(:64572001 ObjectSomeValuesFrom(:609096000 ObjectSomeValuesFrom(:363698007 :64033007))))	Example of EquivalentClasses. which is equivalent to stating that a concept is sufficiently defined by relationships a set of defining relationships. • 90708001 |Kidney disease (disorder)| • 64572001 |Disease| • 609096000 |Role group (attribute)| • 363698007 |Finding site (attribute)| • 64033007 |Kidney structure (body structure)|
900000000000207008 |SNOMED CT core module|	733073007 |OWL axiom reference set|	126516008 |Neoplasm of skin of upper limb (disorder)|	EquivalentClasses(:126516008 ObjectIntersectionOf(:64572001 ObjectSomeValuesFrom(:609096000 ObjectIntersectionOf(ObjectSomeValuesFrom(:116676008 :108369006) ObjectSomeValuesFrom(:363698007 :371311000)))))	Example of a role group with a conjunction of two relationships as its value. • 126516008 |Neoplasm of skin of upper limb (disorder)| • 64572001 |Disease| • 609096000 |Role group (attribute)| • 116676008 |Associated morphology (attribute)| • 108369006 |Neoplasm (morphologic abnormality)| • 363698007 |Finding site (attribute)| • 371311000 |Skin structure of upper limb (body structure)|
900000000000012004 |SNOMED CT model component module|	733073007 |OWL axiom reference set|	774081006 |Proper part of (attribute)|	TransitiveObjectProperty(:774081006)	Example of a transitive object property. • 774081006 |Proper part of (attribute)|
900000000000012004 |SNOMED CT model component module|	733073007 |OWL axiom reference set|	733930001 |Regional part of (attribute)|	SubObjectPropertyOf(ObjectPropertyChain(:127489000 :738774007) :127489000))	Example of a property chain. • 127489000 |Has active ingredient (attribute)| • 738774007 |Is modification of (attribute)|

900000000000012004 |SNOMED CT model component module|	733073007 |OWL axiom reference set|	733929006 |General concept inclusion axiom|	EquivalentClasses(ObjectIntersectionOf(:244066003 ObjectSomeValuesFrom(:733930001 ObjectIntersectionOf(:244066003 ObjectSomeValuesFrom(:733931002 :302548004)))) ObjectIntersectionOf(:244066003 ObjectSomeValuesFrom(:733931002 ObjectSomeValuesFrom(:733930001 :302548004))))	Example of a general concept inclusion (GCI). • 733929006 |General concept inclusion axiom| • 244066003 |Entire skin region (body structure)| • 733930001 |Regional part of (attribute)| • 244066003 |Entire skin region (body structure)| • 733931002 |Constitutional part of (attribute)| • 302548004 |Entire head (body structure)|

Related Links

- SNOMED CT OWL Guide
- SNOMED CT Logic Profile Specification

{

6 SNOMED CT Identifiers

SNOMED Clinical Terms Components are identified and referenced using numeric identifiers. These identifiers have the data type SCTID (SNOMED CT Identifier).

The SCTID data type is 64-bit integer which is allocated and represented in accordance with a set of rules. These rules enable each SCTID to refer unambiguously to a unique component. They also support separate partitions for allocation of Identifiers for particular types of component. In the case of components that originate in an Extension, the SCTID also supports separate namespaces that distinguish between different issuing organizations.

Details of the SCTID are described in the following sections:

6.1 SCTID Data Type

The SCTID data type is a 64-bit positive integer.

When rendered as a string an SCTID must always be represented using decimal digits and when rendered as a string has a maximum permitted length of 18 digits and a minimum length of 6 digits.

Note: Leading zeros are always omitted from the string rendering of an SCTID. For example the value "101291009" must not be rendered as "0101291009".

6.2 SCTID Representation

Each SCTID identifies a SNOMED CT component. The identifier itself does not contain information related to the meaning of a concept or description. This means it is not possible to infer anything about the meaning of a concept from the numeric value of the identifier or from the sequence of digits. The meaning of a concept can be determined from relationships to other concepts and from associated descriptions that include human readable terms.

The SCTID does however have a structure which includes valuable information about the nature and source of the identified component and the validity of the identifier. This structure supports the following features:

- Check-digit validation of the identifier.
    - The check-digit is the final digit in the decimal rendering of the identifier. This can be checked to minimize errors from transcription or incomplete copy-paste actions.
- Partitioning between identifiers for different types of SNOMED CT component.
    - A two-digit partition identifier distinguishes the identifiers of different component types and prevents the same identifier from being allocated to both a concept and a description. As a result, when an SCTID is read from a record or other resource, it is possible to determine whether it represents a concept, a relationship or a description, before searching for the identified component.
- Namespaces to separate component identifiers originated by different organizations.
    - Organizations are only permitted to issue identifiers which fall within a specified namespace of potential identifier values. This prevents collisions between identifiers issues by different organizations which would otherwise result in ambiguity and errors when sharing data.
    - There are two formats used for representing namespaces.
        - Short format in which partition identifiers are reserved for an organization which is permitted to issue any valid identifiers within the allocated partitions. The short format approach does not require a specific namespace identifier and is only applicable to components originated and maintained by the SNOMED International as part of the International Release of SNOMED CT.
        - Long format in which the partition identifier value indicates that a separate namespace identifier is required to distinguish between components originated as part of an extension created by an appropriately authorized organization .

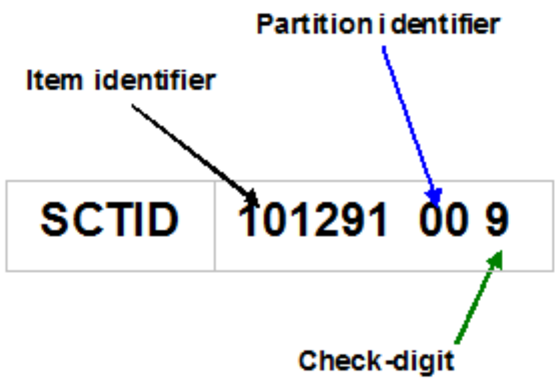


Figure 6.2-1: SCTID Short Format - Applicable to components originating from the International Release

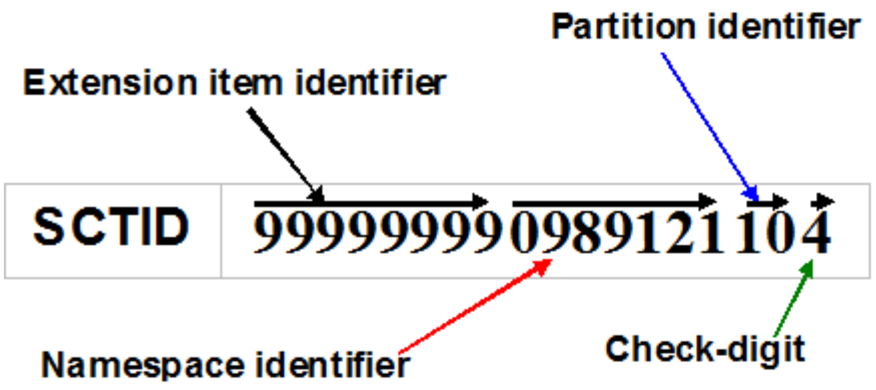


Figure 6.2-2: SCTID Long Format - Applicable to components originating from a SNOMED CT Extension

ⓘ The SNOMED International allocates namespace identifiers to organizations such as Members and Affiliates to enable them to create content and or derivatives in an extension. The namespace identifiers enables unique SCTID to be issued by many organizations and allow each SCTID to be traced to an authorized originating organization.

6.3 SCTID Constraints

The permissible value for the SCTIDs are limited by the following rules:

- Only positive integer values that are greater than 10^5 and less than 10^{18} are permitted.

- The only valid string renderings of the identifier value are String of decimal digits (0-9), commencing with a non zero digit.
- The second and third digits from the right hand end of the string rendering of the identifier must match one of the partition-identifier values specified in this guide.
- The rightmost digit of the string rendering is a check-digit and must match the value calculated using the specified check-digit computation.

Note

- As a result of these rules, many 64-bit integers are not valid SCTIDs. The value limitations enable any valid SCTID to be stored in either a signed or unsigned 64-bit integer.
- The rules also ensure that an SCTID can be distinguished from code from one of the antecedent code systems Read Codes(which are 4 or 5 characters in length) and legacy Identifiers from SNOMED RT and it predecessors (which always start with a letter).
- SNOMED RT identifiers are SCTIDs identical to those used in SNOMED CT but in some cases will now refer to inactive concepts. In these cases, data in the 900000000000489007 |Concept inactivation indicator reference set| and 900000000000522004 |Historical association reference set| can be used to find the identifier of the closest equivalent active concept.

Warning
In some systems and frameworks the default numeric data type is a floating point representation according to IEEE 754. For example, as of today in JavaScript the number type (by default) is a double precision floating precision data type allowing representation of 53-bit integers, which is not sufficient for SCTIDs.
It is recommended that you specify a 64-bit integer type for representing SCTIDs rather than using a default numeric data type. Where a 64-bit integer data type is not available, please use a string data type to represent the SCTIDs.

6.4 Check-digit

The final digit of the SCTID is a check-digit.

Users should be required to type SCTID values but in some case during design and development it may be necessary to copy or paste identifiers. The objective of the check-digit is to detect the commonest types of error that may occur due to typographical errors on those situations or in other cases where transcription or communication mechanisms may introduce error. Examples may include high-level development such as creating or modifying protocols or pre-specified queries.

An SCTID is checked by using the Verhoeff check, which is a Dihedral D 5 Check. This detects a higher proportion of common typographical errors than either the IBM or Modulus 11 check. Unlike the Modulus 11 check it is effective on decimal strings longer than ten-digits. Furthermore its value can always be represented as a decimal digit without excluding any values.

Related Links

- See Check-Digit Computation for detailed information about the Verhoeff check-digit algorithm and links to sample program code.
- See http://snomed.org/verhoeff for a sample web form that can be used to compute a check-digit or check the validity of an SCTID.

6.4.1 SNOMED CT Identifier Check

The form below performs the SCTID Check-Digit computation and checking. It also identifies the namespace element of an identifier. Below the form is an expandable box including the JavaScript code used to perform these computations.

Partial Identifier (without check-digit)		Compute
SNOMED CT Identifier		Check
Result of check		
Component type		
Namespace		

This Verhoeff checking part of this code was based on a webpage at:

- http://www.augustana.ab.ca/~mohrj/algorithms/checkdigit.html

The source HTML and JavaScript code for this form is shown in 6.4.2 Check-digit Computation.

6.4.2 Check-digit Computation

The SCTID(see 3.1.4.2. Component features - Identifiers) includes a check-digit, which is generated using Verhoeff's dihedral check. This section explains the algorithm used and includes sample source code for generating and checking the check-digit in Java Script and Microsoft Visual Basic.

Verhoeff's Dihedral Group D5 Check

The mathematical description of this technique may appear complex but in practice it can be reduced to a pair of two-dimensional arrays, a single dimensional inverse array and a simple computational procedure. These three arrays are shown in the following tables.

- The first array contains the result of "Dihedral D5" multiplication;
- The second array consists of 8 rows of which two are defined while the rest are derived by applying the following formula: *F(i, j) = F(i - 1, F(1, j))* ;
- The third array consists of a single row containing the inverse of the Dihedral D5 array it identifies the location of all the zero values in the first array.

Table 6.4.2-3: Results of Dihedral D5 multiplication

	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
0	0	1	2	3	4	5	6	7	8	9
1	1	2	3	4	0	6	7	8	9	5
2	2	3	4	0	1	7	8	9	5	6
3	3	4	0	1	2	8	9	5	6	7
4	4	0	1	2	3	9	5	6	7	8
5	5	9	8	7	6	0	4	3	2	1
6	6	5	9	8	7	1	0	4	3	2

	0	1	2	3	4	5	6	7	8	9
7	7	6	5	9	8	2	1	0	4	3
8	8	7	6	5	9	3	2	1	0	4
9	9	8	7	6	5	4	3	2	1	0

Table 6.4.2-3: The full array for Function F

	0	1	2	3	4	5	6	7	8	9
0	0	1	2	3	4	5	6	7	8	9
1	1	5	7	6	2	8	3	0	9	4
2	5	8	0	3	7	9	6	1	4	2
3	8	9	1	6	0	4	3	5	2	7
4	9	4	5	3	1	2	6	8	7	0
5	4	2	8	6	5	7	3	9	0	1
6	2	7	9	3	8	0	6	4	1	5
7	7	0	4	6	9	1	3	2	5	8

Table 6.4.2-3: The Inverse D5 array

0	1	2	3	4	5	6	7	8	9	
0	4	3	2	1	5	6	7	8	9	

The identifier is checked by starting at the rightmost digit of the identifier (the check-digit itself) and proceeding to the left processing each digit as follows:

- *Check* = ArrayDihedralD5 (*Check*, ArrayFunctionF((*Position* Modulus 8), *Digit*))

Check = the running value of the check-sum (starts at zero and modified by each step).

Position = the position of the digit (counted from the right starting at zero).

Digit = the value of the digit.

The final value of *Check* should be zero. Otherwise the check has failed.

When calculating the check-digit the same process is applied with a minor variation:

- *Position* is the position that the digit will have when the check-digit has been appended.
- The final value of *Check* is applied to the Inverse D5 array to find the correct check-digit.

Check-digit= ***ArrayInverseD5*** (*Check*).

Sample Java Script for computing Verhoeff's Dihedral Check

A live version of an HTML form and JavaScript is available in section 6.4.1 SNOMED CT Identifier Check.

HTML Code for Form Calling the JavaScript below

```
<style>
p.p1 {margin: 0.0px 0.0px 0.0px 0.0px; font: 12.0px Helvetica}
span.s1 {color: #021da7}
span.s2 {color: #f9975e}
span.s3 {color: #ff9450}
span.s4 {color: #ab4500}
span.s5 {color: #a7a400}
table {border-width: 6px; border-color: #0080ff; border-collapse: collapse; border-
style: ridge;}
td {border-width: 3px; border-color: #0080ff; border-collapse: collapse; padding:
6px; border-style: ridge;}
</style>
<form action="" name="form">
    <table width="441">
        <tr>
            <td width="212" height="25"> Partial Identifier <br/>(without check-
digit)&nbsp; </td>
            <td width="115" height="25">
                <input name="num" size="18"/>
            </td>
            <td width="92" height="25">
                <input onclick="VerhoeffCompute()" type="button" value="Compute"/>
            </td>
        </tr>
        <tr>
            <td width="212" height="35"> SNOMED CT Identifier </td>
            <td width="115" height="35">
                <input name="numcd" size="18"/>
            </td>
            <td width="92" height="35">
                <input onclick="VerhoeffCheck()" type="button" value="Check"/>
            </td>
        </tr>
        <tr>
            <td width="212" height="23"> Result of check&nbsp; </td>
            <td width="115" height="23" colspan="2" id="out"> </td>
        </tr>
        <tr>
            <td width="212" height="23"> Component type </td>
            <td width="115" height="23" colspan="2" id="component"> </td>
        </tr>
        <tr>
            <td width="212" height="23"> Namespace </td>
            <td width="115" height="23" colspan="2" id="extnamespace"> </td>
        </tr>
    </table>
    <p style="margin-left: 0; margin-right: 0"> This Verhoeff checking part of this
code was based
        on a webpage at: </p>
    <ul>
        <li>
```

```
                <a href="http://www.augustana.ab.ca/~mohrj/algorithms/checkdigit.html">
                    http://www.augustana.ab.ca/~mohrj/algorithms/checkdigit.html </a>
            </li>
        </ul>
    </form>
```

Java Script Code for SCTID Validation and Check-Digit Computation

```
var FnF = new Array();
    FnF[0] = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9];
    FnF[1] = [1, 5, 7, 6, 2, 8, 3, 0, 9, 4];
    for ( var i = 2; i < 8; i++ )
    {
        FnF[i] = [,,,,,,,,,];
        for ( var j = 0; j < 10; j++ )
        FnF[i][j] = FnF[i - 1][FnF[1][j]];
    }
var Dihedral = new Array(
   [0, 1, 2, 3, 4, 5, 6, 7, 8, 9],
   [1, 2, 3, 4, 0, 6, 7, 8, 9, 5],
   [2, 3, 4, 0, 1, 7, 8, 9, 5, 6],
   [3, 4, 0, 1, 2, 8, 9, 5, 6, 7],
   [4, 0, 1, 2, 3, 9, 5, 6, 7, 8],
   [5, 9, 8, 7, 6, 0, 4, 3, 2, 1],
   [6, 5, 9, 8, 7, 1, 0, 4, 3, 2],
   [7, 6, 5, 9, 8, 2, 1, 0, 4, 3],
   [8, 7, 6, 5, 9, 3, 2, 1, 0, 4],
   [9, 8, 7, 6, 5, 4, 3, 2, 1, 0] );

var InverseD5 = new Array(0, 4, 3, 2, 1, 5, 6, 7, 8, 9 );

function VerhoeffCheck()

    {
    var check = 0;
    var IdValue = document.form.numcd.value;
    document.getElementById("out").innerText = "";
    document.getElementById("out").setAttribute("style","color:red;");
    document.getElementById("component").innerText ="Invalid partition";
    document.getElementById("component").setAttribute("style","color:green;");
    document.getElementById("extnamespace").innerText ="No namespace";
    document.getElementById("extnamespace").setAttribute("style","color:red;");

    for ( var i=IdValue.length-1; i >=0; i-- )
    check = Dihedral[check][FnF[(IdValue.length-i-1) % 8][IdValue.charAt(i)]];
    if ( check != 0 ) { document.getElementById("out").innerText = "Check-digit
ERROR"; }
    else if  (IdValue.length < 6) {document.getElementById("out").innerText = "SCTID
too short";}
    else if  (IdValue.length > 18) {document.getElementById("out").innerText = "SCTID
too long";}
    else {document.getElementById("out").innerText = "Check-digit OK";
    document.getElementById("out").setAttribute("style","color:green;");
    switch (IdValue.substr(IdValue.length-3,2))
    {
    case "00":
        document.getElementById("component").innerText ="Concept";
        document.getElementById("extnamespace").innerText ="International";
        break;
    case "01":
        document.getElementById("component").innerText ="Description";
        document.getElementById("extnamespace").innerText ="International";
        break;
    case "02":
        document.getElementById("component").innerText ="Relationship";
        document.getElementById("extnamespace").innerText ="International";
        break;
    case "03":
        document.getElementById("component").innerText ="Subset (RF1)";
        document.getElementById("extnamespace").innerText ="International";
        break;
    case "04":
        document.getElementById("component").innerText ="Cross Map Set (RF1)";
        document.getElementById("extnamespace").innerText ="International";
        break;
    case "05":
        document.getElementById("component").innerText ="Cross Map Target (RF1)";
        document.getElementById("extnamespace").innerText ="International";
        break;
    case "10":
        document.getElementById("component").innerText ="Concept";
        document.getElementById("extnamespace").innerText
=IdValue.substr(IdValue.length-10,7);
        break;
    case "11":
        document.getElementById("component").innerText ="Description";
        document.getElementById("extnamespace").innerText
=IdValue.substr(IdValue.length-10,7);
        break;
    case "12":
        document.getElementById("component").innerText ="Relationship";
        document.getElementById("extnamespace").innerText
=IdValue.substr(IdValue.length-10,7);
        break;
    case "13":
        document.getElementById("component").innerText ="Subset (RF1)";
        document.getElementById("extnamespace").innerText
=IdValue.substr(IdValue.length-10,7);
        break;
    case "14":
        document.getElementById("component").innerText ="Cross Map Set (RF1)";
        document.getElementById("extnamespace").innerText
=IdValue.substr(IdValue.length-10,7);
        break;
    case "15":
        document.getElementById("component").innerText ="Cross Map Target (RF1)";
        document.getElementById("extnamespace").innerText
=IdValue.substr(IdValue.length-10,7);
        break;
    default:
        document.getElementById("component").setAttribute("style","color:red;");
    }
```

```
    if (document.getElementById("extnamespace").innerText=='International')
{document.getElementById("extnamespace").setAttribute("style","color:green;");}
    else if (IdValue.length>10) {document.getElementById("extnamespace").setAttribute
("style","color:green;");}
    else  {document.getElementById("extnamespace").innerText="Invalid Namespace";
    }
    }
    }
function VerhoeffCompute( )

    {
    var IdValue = document.form.num.value; var check = 0;
    document.form.numcd.value= "";
    for ( var i = IdValue.length-1; i >=0; i-- )
    check = Dihedral[check][FnF[(IdValue.length-i) % 8][IdValue.charAt(i)]];
    document.form.numcd.value = document.form.num.value + InverseD5[check];
    VerhoeffCheck();
    document.getElementById("out").innerText = "Computed check-digit";
    }
```

Sample Visual Basic for computing Verhoeff's Dihedral Check

Visual Basic Code for Check-Digit Computation

```
Option Explicit
Private Dihedral(9) As Variant
Private FnF(7) As Variant
Private InverseD5 As Variant
Public Function VerhoeffCheck(ByVal IdValue As String) As Boolean
'Check the supplied value and return true or false
Dim tCheck As Integer, i As Integer
VerhoeffArrayInit
For i = Len(IdValue) To 1 Step -1
tCheck = Dihedral(tCheck)(FnF((Len(IdValue) - i) Mod 8)(Val(Mid(IdValue, i, 1))))
Next
VerhoeffCheck = tCheck = 0
End Function
Public Function VerhoeffCompute(ByVal IdValue As String) As String
'Compute the check digit and return the identifier complete with check-digit
Dim tCheck As Integer, i As Integer
VerhoeffArrayInit
For i = Len(IdValue) To 1 Step -1
tCheck = Dihedral(tCheck)(FnF((Len(IdValue) - i + 1) Mod 8)(Val(Mid(IdValue, i, 1))))
Next
VerhoeffCompute = IdValue &amp; InverseD5(tCheck)
End Function
Private Sub VerhoeffArrayInit()
'Create the arrays required
Dim i As Integer, j As Integer
'if already created exit here
If VarType(InverseD5) >= vbArray Then Exit Sub
'create the DihedralD5 array
Dihedral(0) = Array(0, 1, 2, 3, 4, 5, 6, 7, 8, 9)
Dihedral(1) = Array(1, 2, 3, 4, 0, 6, 7, 8, 9, 5)
Dihedral(2) = Array(2, 3, 4, 0, 1, 7, 8, 9, 5, 6)
Dihedral(3) = Array(3, 4, 0, 1, 2, 8, 9, 5, 6, 7)
Dihedral(4) = Array(4, 0, 1, 2, 3, 9, 5, 6, 7, 8)
Dihedral(5) = Array(5, 9, 8, 7, 6, 0, 4, 3, 2, 1)
Dihedral(6) = Array(6, 5, 9, 8, 7, 1, 0, 4, 3, 2)
Dihedral(7) = Array(7, 6, 5, 9, 8, 2, 1, 0, 4, 3)
Dihedral(8) = Array(8, 7, 6, 5, 9, 3, 2, 1, 0, 4)
Dihedral(9) = Array(9, 8, 7, 6, 5, 4, 3, 2, 1, 0)
'create the FunctionF array
FnF(0) = Array(0, 1, 2, 3, 4, 5, 6, 7, 8, 9)
FnF(1) = Array(1, 5, 7, 6, 2, 8, 3, 0, 9, 4)
'compute the rest of the FunctionF array
For i = 2 To 7
FnF (i) = Array(0, 0, 0, 0, 0, 0, 0, 0, 0, 0)
For j = 0 To 9
FnF (i)(j) = FnF(i - 1)(FnF(1)(j))
Next
Next
```

```
'Create the InverseD5 array
InverseD5 = Array("0", "4", "3", "2", "1", "5", "6", "7", "8", "9")
End Sub
```

Reasons for using a check-digit

Although a user should rarely type the SCTID, experience suggests that from time to time this will happen. A user may also copy and paste an SCTID. There is a significant risk of errors in these processes and inclusion of a check-digit is intended to reduce the risk of such errors passing undetected. The choice of check-digit algorithm has been made to maximize the detection of common typographical errors. These have been analyzed by in a paper by J. Verhoeff ("Error Detecting Decimal Codes", *Mathematical Center Tract 29*, The Mathematical Center, Amsterdam, 1969) and subsequently cited in Wagner and Putter, ("Error Detecting Decimal Digits", *CACM*, Vol 32, No. 1, January 1989). These papers give a detailed categorization of the sorts of errors humans make in dealing with decimal numbers, based on a study of 12000 errors:

- single errors: a becomes b (60% to 95% of all errors).
- omitting or adding a digit (10% to 20%).
- adjacent transpositions: ab becomes ba (10% to 20%).
- twin errors: aa becomes bb (0.5% to 1.5%).
- jump transpositions: acb becomes bca (0.5% to 1.5%).
- jump twin errors: aca becomes bcb (below 1%).
- phonetic errors: a0 becomes 1a -similar pronunciation e.g. thirty or thirteen (0.5% to 1.5%).

In the explanations above, a is not equal to b, but c can be any decimal digit.

A brief comparison of check-digit effectiveness

The IBM Check

The check-sums used for credit cards (the IBM check) picks up the most common errors but miss some adjacent transpositions and many jump transpositions. Assuming the pattern of errors described above, on average it will miss between 4% and 5% of expected errors.

The ISBN Check (Modulus 11)

The ISBN modulus 11 (used for UK NHS number) picks up more errors than the IBM checksum. Leaving 2% to 3% of errors undetected. However, it generates a check-sum value of 0 to 10 and thus cannot be represented as a single check-digit in about 9% of cases. The ISBN convention is to use "X" to represent the check-digit value 10 but this is incompatible with an Integer representation. The UK NHS number uses this check-sum but regards and number generating a check-sum of 10 as an invalid identifier. This approach could be applied to the SCTID but this would render 9% of possible values unusable in each partition and namespace. This would prevent a simple sequence of values from being allocated as the *item identifier* within any namespace. More significantly the unusable *item identifier* would differ in each namespace or partition and this would prevent simple transpositions of *item identifiers* between partitions and namespaces.

Partitions could be a useful way of distinguishing developmental and released components and revising the partition and recalculating the check-digit would then be an elegant way to activate these components for a distribution version. It seems unwise to prevent future development and maintenance by using a check-sum that will prevent this.

Verhoeff's Check

Verhoeff's check catches all single errors, all adjacent transpositions, over 95% of twin errors, over 94% of jump transpositions and jump twin errors, and most phonetic errors. Therefore, like modulus 11, the Verhoeff check reduces the undetected error rate to 2% or 3%. Unlike modulus 11, it does this using a single decimal check-digit and without limiting the range of valid numbers.

The majority of the undetected errors with both modulus 11 and Verhoeff result from additions or omissions of digits. Any check-digit method is likely to miss 10% of such errors and since these comprise 10% to 20%. The Verhoeff scheme also misses four jump twin errors involving digits with a difference of 5 (i.e. 050 vs. 505, 161 vs. 616, 272 vs. 727, and 494 vs. 949).

6.5 Partition Identifier

The penultimate two-digits of the SCTID (second and third from the right), are the partition identifier.

The partition identifier indicates the nature of the component identified. This allows the identifier of a description to be distinguished from the identifier of a concept.

The partition identifier also indicates whether the SCTID contains a namespace identifier (*long format*) or follows the *short format* applicable to identifiers of components that originated in the International Release. Identifiers of components that originated in the International Release of SNOMED CT have one of the following partition identifier values:

Table 6.5-2: Partition identifier Values for Short Format SCTIDs

PartitionId	Component type
00	Concept
01	Description
02	Relationship

Identifiers of components that originated in an extension have one of the following partition identifier values:

Table 6.5-2: Partition identifier Values for Long Format SCTIDs

PartitionId	Component type
10	Concept
11	Description
12	Relationship

All other partition identifier values are reserved for future use.

6.6 Namespace-Identifier

If the partition-identifier indicates a long format SCTID, the seven-digits immediately to the left of the partition-digit are a namespace-identifier. The namespace-identifier is an integer value, left padded with 0 s as necessary to ensure there are always seven digits in the value. The namespace-identifier does not hold meaning.

Each organization that is authorized to generate SCTID is allocated a namespace-identifier by the SNOMED International. Each allocated namespace is represented in the Namespace Concept metadata sub-hierarchy, released as part of the International release (see details in The Namespace hierarchy).

6.7 Item-Identifier Digits

The string of digits to the left of the partition-identifier(in a *short format* SCTID) or to the left of the namespace-identifier(in a *long format* SCTID) is referred to as the *item-identifier*.

These values are available to uniquely identify an individual entity within the specified partition or namespace. The same *item-identifier* can be allocated in each partition of each namespace as the SCTID is rendered unique by the partition-identifier and the namespace-identifier. For components in the International Release of SNOMED CT, *item-identifiers* will usually be issued in the arbitrary order in which components are added to SNOMED Clinical Terms. However, due to management of the editing process the sequence of issued *item-identifiers* may be discontinuous.

CAUTION:

In all cases, the value of an *item-identifier* on its own is meaningless. The only way to determine the meaning of an SCTID is by looking up the complete value in an appropriate distribution file.

6.8 Example SNOMED CT identifiers

The following examples conform to the SNOMED CT identifier specification and illustrate a range of possible Identifiers within different partitions and namespaces.

SctId	partition identifier	check-digit	**Notes**
100005	00 = concept, using short format	5	The Item identifier digits 100 are the lowest permitted value. Therefore this is the lowest SctId that can be allocated to a concept.
100014	01= description, using short format	4	This is the lowest SctId that can be allocated to a description.
100022	02= relationship, using short format	2	This is the lowest SctId that can be allocated to a relationship.
1290023401004	00= concept, using short format	4	A valid SctId for a concept.
1290023401015	01= description, using short format	5	A valid SctId for a description.
9940000001029	02= relationship, using short format	9	A valid SctId for a relationship.
11000001102	10= concept, using long format	2	A valid long format SctId for a concept in the 1000001 namespace.
10989121108	10= concept, using long format	8	A valid long format SctId for a concept in the 0989121 namespace.
1290989121103	10= concept, using long format	3	A valid long format SctId for a concept in the 0989121 namespace.
1290000001117	11= description, using long format	7	A valid long format SctId for a description in the 0000001 namespace.
9940000001126	12= relationship, using long format	6	A valid long format SctId for a relationship in the 0000001 namespace.
999999990989121104	10= concept, using long format	4	The maximum valid SctId for a concept in the 0989121 namespace.

6.9 The Namespace Hierarchy

SNOMED CT core release files include metadata concepts that represent each of the allocated namespace-identifiers.

A namespace concept has the following characteristics:

- It is a subtype child of 370136006 |Namespace concept| .
- The fully specified name of the concept has the term
  - **Extension Namespace** *{nnnnnnn}* **(namespace concept)**
- A synonym associated with each concept has the term
  - **Extension Namespace** *{nnnnnnn}*
- Where appropriate further synonyms may be included to identify the nature of the responsible organization.

In the terms shown above *{nnnnnnn}* is the seven digit namespace identifier of the responsible organization.

Appendix A: Notes on Release File Changes

Representation of the Logical Model - Before July 2018

This is a historical record of the representation of the SNOMED CT Logical Model before July 2018. During a period of transition between July 2018 and July 2019 the model was revised to enable more advanced description logic axioms to be represented. The Logical model following those changes is shown in 2.2 Representation of the Logical Model

Figure 1 shows how SNOMED CT release files prior to July 2018 represented the various elements in the logical model shown on Figure 1-1: High-level abstract view of the design of SNOMED CT. Further details are provided by Table 1.

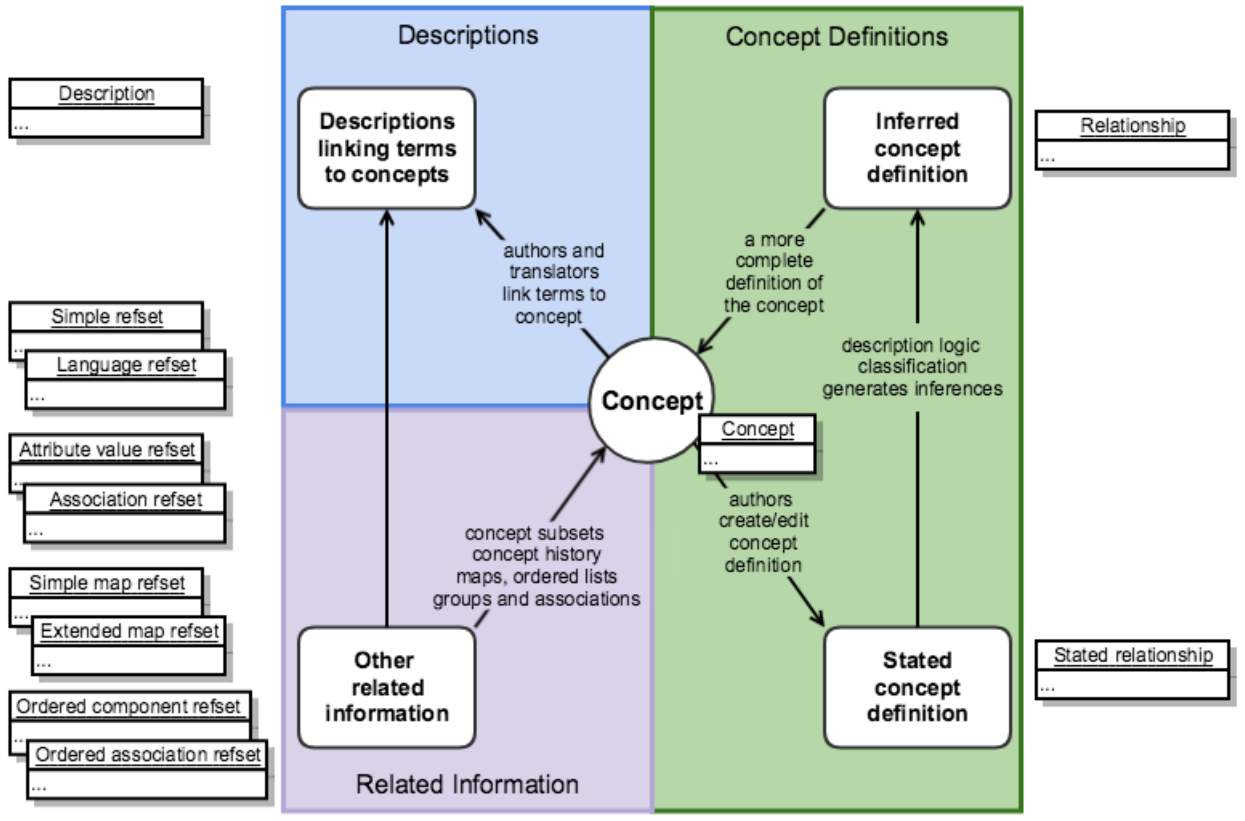

Descriptions
Concept Definitions
Description
...
Descriptions linking terms to concepts
authors and translators link terms to concept
Inferred concept definition
a more complete definition of the concept
Relationship
...
Simple refset
Language refset
Attribute value refset
Association refset
Simple map refset
Extended map refset
Ordered component refset
Ordered association refset
Concept
description logic classification generates inferences
Concept
...
authors create/edit concept definition
concept subsets concept history maps, ordered lists groups and associations
Other related information
Stated concept definition
Stated relationship
...
Related Information

Figure 1: Practical representation of the logical model of SNOMED CT (before update)

Table 1: Release file representation of the logical model (before update)

Logical Model	Release File Representation	References
Concepts	Each concept is represented by a row in the concept release file.	4.2.1 Concept File Specification
Descriptions	Each description is represented by a row in the description release file.	4.2.2 Description File Specification

Stated Concept Definitions	Each stated concept definition is represented by a set of rows in the stated relationship release file. Each row in the set that set defines a concept, represents a defining relationship with another concept. The definitionStatusId column in the concept file row indicates whether the set of defining relationships is sufficient to define the concept. The stated relationship file has the same format as the relationship file. ⚠ This representation of stated definitions is being replaced by a transition process starting in July 2018 and ending during 2019.	4.2.3 Relationship File Specification Stated Relationship File
Inferred Concept Definitions	Each inferred concept definition is represented by a set of rows in the relationship release file. Each row in the set that set defines a concept, represents a defining relationship with another concept. The definitionStatusId column in the concept file row indicates whether the set of defining relationships is sufficient to define the concept.	4.2.3 Relationship File Specification
Other Related Information	Represented by a range of reference set release files that conform to the extensible reference set file format. Each row in a reference set refers to a concept or description as a member of the set.The extensible structure allows different types of related information to be associated with the referenced component.	5.2 Reference Set Types Practical Guide to Reference Sets

Associations Between Release Files Prior to July 2018

⚠ For details of the Stated Definition View since July 2019 please see 4.1 Associations Between Release Files.

The stated relationship file, was used to represent the stated view of relationships prior to a transition process between July 2018 and July 2019. In July 2019 this file became obsolete and was completely replaced by two OWL Expression Reference Sets (the OWL axiom reference set file and the OWL ontology reference set file).

Figure 1 shows the associations between files that represent the stated and inferred views prior to the transition. Figure 4.1-2 shows the associations between the release files that following these changes.

The stated relationship file is now obsolete as it has been completely replaced by two OWL Expression Reference Sets (the OWL axiom reference set file and the OWL ontology reference set file).

During the transitional period the stated relationship file continued to be distributed, but the OWL axiom reference set introduced advanced aspects to concept definitions which could not be represented in the stated relationship file.

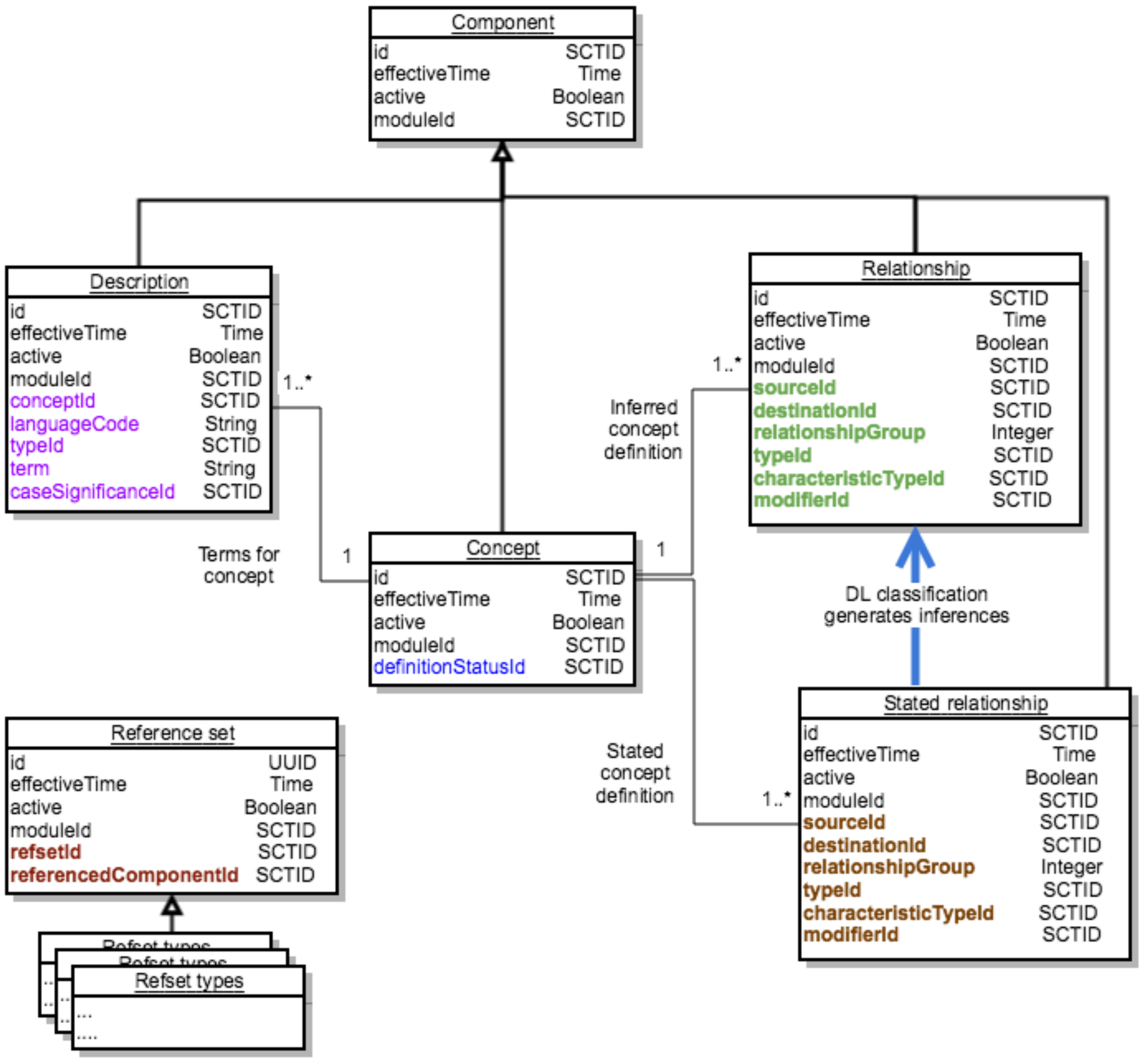


Figure 1: Associations between SNOMED CT Release Files Prior to July 2019 (now obsolete)

Notes on modifierId

These notes provide additional information about the modifierId column in the Relationship File. The modifierId column was included in the specification of the relationships file in the expectation that it would in future distinguish between different types of description logic axiom. However, in practice it has not been used. Different approaches to enhanced use of description logic are under consideration and it now seems unlikely that this column will be used as originally intended. Therefore, until further notice it is recommended that the contents of this column should be ignored.

The following notes were included in the original specification of modifierId and are retained here for consistency.

Original release notes on modifierId

The modifierId field will initially be set to 900000000000451002 |Some| to keep compatibility with the RF1 release. Widening the range of this field to include other values (such as |All|) will in future increase the expressive power of SNOMED CT. However, this is likely to come at the cost of an increase in reasoning complexity, leading to potential issues for classification tooling.

Notes:

1. The modiferId field has been included at this stage as the RF2 format is likely to be stable for at least a five year period, without addition or deletion of fields. Within that period it is anticipated that other modifierId values will be added. Therefore, although not fully implemented at this stage, this field has been included in the initial RF2 specification as it represents an integral part of the Description Logic used by SNOMED CT.
2. Any expansion of SNOMED CT to include relationships with a modifierId set to a value other than 900000000000451002 |Some| will be discussed with Members prior to introduction.
3. Changes have been made to the "Immutability" values shown in the above table in the 2014-07-31 version. These changes reflect the fact that the values in the following columns of a uniquely identified relationship have occurred in historical data and in these cases tracking the history of these changes is of greater value that insisting on immutability.
    - relationshipGroup: The number can change though the logical content of the group represented should not change. Additionally no significance should be read into the relationshipGroup value of an inactive relationship;
    - characteristicType: This has changed in historical data but should not change in future;
    - modifierId: Since there is currently only one value for this no changes are possible but if the permitted values are extended as suggested above then it is likely that changes would be required.

RF1 Compatibility and Conversion Tools

In January 2012 the SNOMED International switched from the original Release Format(used for SNOMED CT distribution since 2002), to the more flexible and consistent Release Format 2(RF2). This means that from that date onward the primary source data for the SNOMED CT International Release is maintained and distributed in the RF2 format.

The SNOMED International recognizes that, while implementers will which to benefit from the features of the new format, there is inevitably a transitional period during which both format are in use. Therefore, the SNOMED International provides the following resources to support users whose system do not yet support SNOMED CT Release Format 2:

- Release Format 1 files will continue to be included in the International Release for a limited period
    - These files are not the authoritative version of SNOMED CT but are generated from the authoritative RF2 data using a software utility developed for this purpose.
    - The resulting RF1 data retains the functionality of the original release data but does not support any of the features of RF2. While all the clinically relevant SNOMED CT hierarchies are identical in both releases, the additional "Metadata Hierarchy" added as part of the RF2 upgrade is not included in the RF1 converted data. In addition there are some cases where Cross Maps
- The RF2 to RF1 Conversion Tool used for generating the RF1 files is also available to all Members and Affiliate Licensees
    - The "RF2 Conversion Tool" is an open source, Java-based, software tool to facilitate the conversion of SNOMED CT files released in RF2 format into RF1 format. The tool provides both a command line utility and a Graphical User Interface(GUI) to facilitate configuration, progress tracking and the maintenance of additional data whenever it is not available as part of an RF2 release.
    - The limitations of RF2 to RF1 conversion (noted above) will also apply to conversion undertaken using this tool. To enable the conversion to be completed successfully in a way that retains and replaces Identifiers consistently for the RF1 environment a set of auxiliary files (the "RF1 Compatibility Package") is also required.

The "RF2 to RF1 Conversion Tool" and the "RF1 Compatibility Package" are available for Members and Affiliates to download in the same way as the SNOMED CT International Release.

Caution!
These resources and tools are intended for use during a transitional period and should not be considered as a long term alternative to migration to support direct use of RF2 data within applications. As SNOMED CT continues to evolve more of the specific feature of RF2 will be used to add value to the terminology. Some of the added value delivered by RF2 is soon likely to be regarded as essential for effective solutions to user requirements.

Appendix B. Specification Reference Information

This section lists the file and field names used in technical specifications within this guide. The scope of use of these names is limited to the tables in which they are used and the given definitions are not intended for use in any other context.

A

acceptabilityId (field)

A field in a 900000000000506000 |Language type reference set| that indicates the acceptability of a Description in the language or dialect specified by that Reference Set . Values include "preferred" and "acceptable".

Note: Field name in a 900000000000506000 |Language type reference set|

active (field)

A Boolean field that specifies whether an identified component or is an active from the point in time specified by the effectiveTime .

Note: Field name in SNOMED CT Release Format 2.

Related Links

- Meaning of the active field
- 3.2 Release Types
- AAA

alternateIdentifier (field)

A field in the Identifier file containing the representation of an Identifier in another code system that is irrevocably linked to a SNOMED CT identifier .

Related Links

- Identifier

annotation (field)

An Annotation Reference Set field containing additional information linked to a SNOMED CT component .

Note: Field name in SNOMED CT Release Format 2.

Related Links

- 5.2.7 Annotation Reference Set

attributeDescription (field)

A reference to a concept that specifies the name and/or usage of an additional attribute in a Refset. If the attributeType is component reference, the values applied to this additional attribute are restricted to subtypes of this concept .

Note: Field name in a SNOMED CT Release Format 2 Reference Set Descriptor.

attributeOrder (field)

An integer representing the position of an additional attribute in a Refset. The value 0 (zero) refers to the referencedComponentId. All other values refer to the position of an additional attribute relative to the referencedComponentId .

Note: Field name in a SNOMED CT Release Format 2 Reference Set Descriptor.

attributeType (field)

A reference to a concept that specifies the data type of an additional attribute in a Refset .

Note: Field name in a SNOMED CT Release Format 2 Reference Set Descriptor.

B

Boolean (data type)

A datatype that represents either true or false.

Note: In SNOMED CT release files the value 0 (zero) represents "false" and the value 1 (one) represents true.

C

caseSignificanceId (field)

A field in the Description Release File containing a SNOMED CT identifier that indicates whether the text of the term can be modified to by switching characters from upper to lower case (or vice-versa).

Note: Field name in SNOMED CT Release Format 2

Related Links

- 4.1.4 Concept Enumerations for caseSignificanceId
- Description

characteristicTypeId (field)

A reference to a concept that specifies the nature of a Relationship . Values include "defining", "qualifying" etc.

Note: Field name in the SNOMED CT Release Format 2 relationships table.

Concept file

The file structure used to distribute SNOMED CT concepts .

Note: Component File name in SNOMED CT Release Format 2

Related Links

- 4.2.1 Concept File Specification

conceptId (field)

A field in the Description file that associates a term with the concept to which it applies .

Note: Field name in the Description file .

correlationId (field)

A field in the Complex Map Reference Set containing a SNOMED CT identifier which represents the correlation between the SNOMED CT concept and the target code .

Note: Field name in SNOMED CT Release Format 2

Related Links

- Data structure

D

definitionStatusId (field)

A field in the Concept Release File containing a SNOMED CT identifier which specifies whether the concept is fully defined or primitive .

Note: Field name in the SNOMED CT Release Format 2 concepts table.

Related Links

- 4.1.2 Concept Enumerations for definintionStatusId
- Concept

Description file

The file structure used to distribute SNOMED CT descriptions.

Note

- Component File name in SNOMED CT Release Format 2

Related Links

- 4.2.2 Description File Specification

descriptionFormat (field)

A 5.2.13 Description Format Reference Set field reference to a concept that specifies the maximum length and format of the term fields for a particular type of Description .

Note: By default the term is a UTF-8 string of up to 255 characters without markup. However, description types can be specified which are longer in length and/or contain format markup (e.g. HTML). For more details of how this is specified see the file structure specification.

Related Links

- 5.2.13 Description Format Reference Set

descriptionLength (field)

A 5.2.13 Description Format Reference Set field containing an integer which indicates the maximum length of the term string for a specified type of Description .

Note: By default the term is a UTF-8 string of up to 255 characters without markup. However, description types can be specified which are longer in length and/or contain format markup (e.g. HTML). For more details of how this is specified see the file structure specification.

Related Links

- 5.2.13 Description Format Reference Set

destinationId (field)

A field in the Relationship Release File containing a SNOMED CT identifier that refers to the concept that represents the destination (or attribute-value) of the associated Relationship .

Note: Field name in SNOMED CT Release Format 2. In RF1 this field was called *ConceptId2*

Related Links

- Relationship

Dualkey (field)

A key used to facilitate textual searches of SNOMED CT that consists of the first three letters of a pair of words in a Description. All possible pairs of words in each Description may be paired irrespective of their relative position in the Description. *Dualkeys* are represented as a row in the *Dualkeys* Table.

Note: Field name in SNOMED CT toolkit

Dualkey table

A table in which each row represents a Dualkey. See [see 6.1.5.2 Word Search Tables - Summary].

Note: File or Table name in SNOMED CT toolkit

E

effectiveTime (field)

Specifies the inclusive date at which the component version's state became the then current valid state of the component.

Note: Field name in SNOMED CT Release Format 2

Excluded word (field)

A word that in a given language is so frequently used, or has so poor a discriminating power, that it is suggested for exclusion from the indices used to support textual searches of SNOMED CT. *Excluded Words* are represented as a row in the Excluded Words Table

Note: Field name in SNOMED CT toolkit

Excluded words table

A data table in which each row represents an Excluded Word. See [see 6.1.5.2 Word Search Tables - Summary].

Note: File or Table name in SNOMED CT toolkit

I

Identifier file

The file structure used to distribute alternative Identifiers for SNOMED CT components.

Note: The Identifier file is not currently used in the SNOMED CT International Release as use of the more flexible 5.2.9 Simple Map Reference Set structure is preferred for links to alternative codes. The only known current use of this file is for internal identification of components during the content development process.

Related Links

- Identifier file

id (field)

A field that provides the unique identifier of a component (concept, description or relationship) or reference set member .

Note:

- The data type of the *id* for a component is SCTID and this identifier is used to refer to the component .
- The data type of the *id* for a reference set member is UUID. This identifier is only used to support versioning of a rows (member) in a Reference set it does not identify the Reference set itself (see refsetId) nor does it identify to a component refered to by the Reference set (see referencedComponentId).

identifierSchemeId (field)

A field in the RF2 Identifier file containing a SNOMED CT identifier which identifies the alternate code system.

Note: In practice, the identifier file is not used in the SNOMED CT International Release as the use of 5.2.9 Simple Map Reference Set is preferred. The only current use of this file is for internal identification during the development process.

Related Links

- Identifier

Integer (data type)

A datatype that represents a whole number.

Note: In SNOMED CT release file specifications integers are represented as a string of decimal digits. The range of values and support for negative values may be constrained for the specification are specified for each usage of this datatype. However, unless otherwise specified, all release file fields of data type *integer* are assumed to be 32-bit signed integers.

Related Links

- 3.1.2 Release File Data Types

K

Keyword (field)

A field containing a potential search text in one of the WordKey Tables or a word excluded for key generation in the Excluded Words Table .

Note: Field name in SNOMED CT toolkit

L

linkedToId (field)

An Ordered Reference Set field containing a SNOMED CT identifier which refers to either a sub-group of components or a child concept in the alternative hierarchy represented by the Reference set. The parent of grouping component is represented by the referencedComponentId .

Note: Field name in SNOMED CT Release Format 2.

Related Links

- 5.2.20 Ordered Reference Set

M

mapAdvice (field)

Field in a 5.2.10 Complex and Extended Map Reference Sets containing human-readable advice, that may be employed by the software vendor to give an end-user advice on selection of the appropriate target code from the alternatives presented to him within the group.

mapGroup (field)

Field in a 5.2.10 Complex and Extended Map Reference Sets containing an integer that groups a set of complex map records from which one may be selected as a target code. Where a SNOMED CT concept maps onto 'n' target codes , there will be 'n' groups, each containing one or more complex map records.

mapCategoryId (field)

Field in a 5.2.10 Complex and Extended Map Reference Sets that identifies the SNOMED CT concept in the metadata hierarchy which represents the MapCategory for the associated map member.

Note: The categories vary for different target code systems, each set of categories is represented by a subtype of 609331003 |Map category value|. For example in the case of ICD-10 the individual category values are subtypes of:

447634004 |ICD-10 Map category value| .

mapPriority (field)

Field in a 5.2.10 Complex and Extended Map Reference Sets that specifies the order in which complex map records should be checked. Only the first map record meeting the run - time selection criteria will be taken as the target code within each mapGroup .

mapRule (field)

Field in a 5.2.10 Complex and Extended Map Reference Sets containing a machine-readable rule, (evaluating to either 'true' or 'false' at run-time) that indicates whether this map record should be selected within its mapGroup .

mapTarget (field)

Field in a 5.2.9 Simple Map Reference Set or a 5.2.10 Complex and Extended Map Reference Sets that contains the target code(s) to which the SNOMED CT concept represented the referencedComponentId is mapped in the target scheme .

modifierId (field)

A field in the relationship file that indicates the description logic modifier that applies to that defining Relationship (e.g. "some" or "all").

Usage: Field name in SNOMED CT Release Format 2.

moduleId (field)

A field in each component release file which represents the development module within which it was created and is maintained.

Note: Field name in SNOMED CT Release Format 2, which is specified in [see 3.1.6 Module Identification].

O

order (field)

Order ... to be defined.

Note: Field name in SNOMED CT Release Format 2

OWL expression reference set file

A release file that follows the OWL Expression Reference Set pattern and contains expressions that represent general statements about the SNOMED CT ontology and axioms that define SNOMED CT concepts.

Notes

- The OWL expression reference set contains two reference sets, the OWL ontology reference set and the OWL axiom reference set.

Related Links

- OWL ontology reference set
- OWL axiom reference set
- Release File Specification
    - 5.2.21 OWL Expression Reference Set

OWL ontology reference set file

A release file that follows the OWL Expression Reference Set pattern and contains general ontology information related to a SNOMED CT edition.

Q

query (field)

A field in a 5.2.8 Query Specification Reference Set that contains a text string representing criteria for selection of SNOMED CT components to be included in 5.2.1 Simple Reference Set

Note: A standard syntax for use in these queries is currently under development and is due for publication in late 2014.

R

referencedComponentId (field)

A field in a Reference Set containing an Identifier which refers to the component to which a row in the Reference Set applies.

Note: This field is present in all types of Reference Set and, unless otherwise specified, the field data type is SCTID .

Related Links

- 5.2 Reference Set Types
- Simple Reference Set
- The basic reference set member file format

Reference Set file

The file structure used to distribute SNOMED CT Reference sets .

Related Links

- 3.2.1. Reference Sets
- 5.2 Reference Set Types

refsetId (field)

A field in a Reference Set which uniquely Identifier which refers to the component to which a row in the Reference Set applies.

Note: This field is present in all types of Reference Sets and its data type is SCTID. It links together all the members of a Reference Set and refers to a concept that names the Reference Set .

Related Links

- 5.2 Reference Set Types
- Simple Reference Set
- The basic reference set member file format

Relationship file

The file structure used to distribute SNOMED CT relationships.

Related Links

- 4.2.3 Relationship File Specification
- Concept Enumerations for Relationship typeId

relationshipGroup (field)

Field in the Relationship File is used to group Relationships together for a concept. For example, where a particular type of prosthesis is inserted a joint, the Defining characteristics describing the prosthesis type would be in one group whereas those describing the location or laterality of the joint would be in another group.

S

SCTID (data type)

A unique integer identifier applied to each SNOMED CT component (Concept, Description, Relationship).

Note: The value of an SCTID is structured to include an item identifier, a check-digit and a partition identifier. Depending in the value of the partition identifier it may also include a namespace identifier.

Related Links

- 3.1.2 Release File Data Types
- 6 SNOMED CT Identifiers

sourceEffectiveTime (field)

A field in the Module Dependency Reference Set which specifies the effectiveTime of the version of the source module with depends on the specified version of the target module. The effectiveTime must match exactly.

Note: Field name in SNOMED CT Release Format 2

Related Links

- 4.2.4 Module Dependency Reference Set
- 5.2.12 Module Dependency Reference Set

sourceId (field)

A field in the Relationship Release File containing a SNOMED CT identifier that refers to the concept that represents the source of the associated Relationship. The *sourceId* refers to the concept that is defined by the Relationship .

Note: Field name in SNOMED CT Release Format 2. In RF1 this field was called *ConceptId1*

Related Links

- Relationship

Stated Relationship File

A distribution file containing the stated form of SNOMED CT relationships .

Notes:

1. The stated form of a Concept is the Description Logic definition that is directly edited by authors or editors. It consists of the stated 116680003 |is a| relationships plus the defining relationships that exist prior to running a classifier on the logic definitions. Therefore, the stated form of a Concept is represented by a collection of relationships: one or more 116680003 |Is a| relationships and zero or more defining relationships .
2. The Stated Relationships File is in the same table format as the Relationships File, but the value of the characteristicTypeId field is 900000000000010007 |Stated relationship (core metadata concept)| .

String (data type)

A datatype representing a sequence of characters.

Note: In SNOMED CT release file specifications strings are represented using *UnicodeUTF-8* encoding.

Related Links

- 3.1.2 Release File Data Types
- Appendix C. Unicode UTF-8 encoding

T

targetComponentId (field)

An Association Reference Set field containing a SNOMED CT identifier which specifies the target of the association from the source component (e.g. a concept or Description) referred to by the referencedComponentId .

Note: Field name in SNOMED CT Release Format 2.

Related Links

- 5.2.5 Association Reference Set

targetEffectiveTime (field)

A field in the Module Dependency Reference Set which specifies the effectiveTime of the version of the target module on which the specified version of the source module depends. The effectiveTime must match exactly.

Note: Field name in SNOMED CT Release Format 2

Related Links

- 4.2.4 Module Dependency Reference Set
- 5.2.12 Module Dependency Reference Set

term (field)

A text string that represents the concept referenced by the conceptId field in the Description file .

Note:

By default the term is a UTF-8 string of up to 255 characters. However, description types can be specified which are longer in length and/or contain format markup (e.g. HTML).

Field name in the Description file .

Time (data type)

A datatype representing a date or time.

Note: In SNOMED CT release file specifications date and times are represented as strings using the ISO 8601 basic format.

- The date format used is YYYYMMDD.
- Where time is included the format is YYYYMMDDThhmmssZ. The time is separated from the date by the letter "T" and followed by the letter "Z" indicating that the timezone is UTC.

Examples:

July 31st 2012: **20120731** .

13:15 UTC on August 2nd 2012: **20120802T131500Z**

Related Links

- http://en.wikipedia.org/wiki/ISO_8601

- 3.1.2 Release File Data Types

Transitive closure file

A file containing the transitive closure of the SNOMED CT subtype hierarchy.

The transitive closure file is not currently distributed but can be generated from the snapshot relationship file using a script file. The script file is available for download at http://snomed.org/transclose or via GitHub as part of one of the database loaders (mysql-loader-with-optimized-views).

Related Links

- See 4.2.5 Transitive Closure Files for further details.

typeId (field)

A field in the Description and Relationship Release Files which contains a SNOMED CT identifier that represents the type of Description or Relationship represented.

- Description. ***typeId*** represents the type of Description. Description types include subtypes of 900000000000446008 |Description type (core metadata concept)|. These include 900000000000013009 | Synonym (core metadata concept)| and 900000000000003001 |Fully specified name (core metadata concept) |. There is no *typeId* value for " Preferred term " as the preferred term is the synonym marked as "Preferred" in the appropriate [see 4.2.1 Language Reference Sets].
- Relationship. ***typeId*** represents the type of Relationship between the concept identified by sourceId and the concept identified by destinatonId. Relationship types are 116680003 |Is a (attribute)| and subtypes of 410662002 |Concept model attribute (attribute)| .

Note: Field name in the Description file and in the Relationship file .

Related Links

- Concept Enumerations for Description typeId
- Concept Enumerations for Relationship typeId
- Concept model attribute
- Description
- Relationship

U

Unicode

A standard character set, which represents most of the characters used in the world using a 16-bit encoding.

Note: The Unicode character set can be encoded using either UTF-16 or UTF-8. UTF-16 uses two bytes for every character. UTF-8 is able to store the most commonly used characters in western alphabets using a single byte, but it requires two bytes to encode accented characters and three bytes to encode symbols used in many non-European scripts.

UTF-16

A standard method of directly encoding Unicode using two bytes for every character.

Note: SNOMED CT release files do not use UTF-16. However, the UTF-8 representation used in release files can be converted to UTF-16.

Related Links

- 3.1.2 Release File Data Types

- Appendix C. Unicode UTF-8 encoding

UTF-8

A standard method of encoding Unicode characters in a way optimized for the ASCII character set. *UTF-8* is described in [see Appendix C. Unicode UTF-8 encoding].

Note: This encoding is used for release file fields of data type "String".

Related Links

- 3.1.2 Release File Data Types
- Appendix C. Unicode UTF-8 encoding

UUID (data type)

A datatype representing a Universally Unique Identifier encoded as a 128-bit integer.

Note: In SNOMED CT release files *UUIDs* are represented as a string following the standard canonical form. In this string form a *UUID* is represented by 32 hexadecimal digits, displayed in five groups separated by hyphens, in the form 8-4-4-4-12 for a total of 36 characters (32 digits and four hyphens).

Example: ac527bed-9c70-4aad-8fc9-015828b148d9

Related Links

- http://en.wikipedia.org/wiki/Universally_unique_identifier

V

valueId (field)

ValueId ... to be defined.

Note: Field name in SNOMED CT Release Format 2

W

Word equivalents table

A data table in which each row represents a Word Equivalent. See [see 6.1.3 Word Equivalents].

Note: File or Table name in SNOMED CT toolkit

WordBlockNumber (field)

A field in the Word Equivalents Table , which links together several rows which have an identical or similar meaning.

Note: Field name in SNOMED CT toolkit

WordKey table

A data table relating each word used in SNOMED CT (other than Excluded Words) to the Descriptions. See [see 6.1.5.2 Word Search Tables - Summary].

Note: File or Table name in SNOMED CT toolkit

WordRole (field)

A field in the Word Equivalents Table, which specifies the usual usage of this word, abbreviation or phrase, or the usage in which it has a similar meaning to the text in one or more other rows of the table that share a common WordBlockNumber .

Note: Field name in SNOMED CT toolkit

WordText (field)

A field in the Word Equivalents Table, which contains a word, phrase, acronym or abbreviation that is considered to be similar in meaning to the text in one or more other rows of the table that share a common WordBlockNumber .

Note: Field name in SNOMED CT toolkit

WordType (field)

A field in the Word Equivalents Table , which specifies whether this row contains a word, phrase, acronym or abbreviation.

Note: Field name in SNOMED CT toolkit

Appendix C. Unicode UTF-8 encoding

UTF-8 is an efficient encoding of Unicode character - String that recognizes the fact that the majority of text-based communications are in ASCII. It therefore optimizes the encoding of these characters.

Unicode is preferred to ASCII because it permits the inclusion of accents, scientific symbols and characters used in languages other than English. The UTF-8 format is a standard encoding that provides the most efficient means of encoding 16-bit Unicode characters in cases where the majority of characters are in the ASCII range. Both UTF-8 and the alternative UTF-16 encoding are supported by all widely used operating systems and major applications. UTF-8 was adopted is an IETF Internet Standard (it was initially adopted by IETF in 1996 to restrict some code values in 1998 and 2003). In 2008 UTF-8 became the most widely used for of encoding in web pages.

SNOMED CT uses the UTF-8 representation[1] of characters in terms and other text fields.

---

[1] Note that SNOMED CT does not use, or require use of, the Byte Order Mark (BOM) specified by the Unicode standard because all SNOMED CT release files use UTF-8.

Summary of Unicode Encoding Rules

Character encoding

- ASCII characters (in the range 0-127) are encoded as a single byte.
- Greek, Hebrew, Arabic and most accented European characters are encoded as two bytes;
- Other characters are encoded as three bytes;
- The individual characters are encoded according to the following rules.

Single byte encoding

Characters in the range 'u+0000' to 'u+007f' are encoded as a single byte.

Table -5: UTF-8 Single Byte Encoding

byte 0	
0	bits 0-6

Two byte encoding

Characters in the range 'u+0080' to 'u+07ff' are encoded as two bytes.

Table -5: Two byte encoding

byte 0				byte 1		
1	1	0	bits 6-10	1	0	bits 0-5

Three byte encoding

Characters in the range 'u+0800' to 'u+ffff' are encoded as three bytes:

Table -5: UTF-8 Three Byte Encoding

byte 0					**byte 1**			**byte 2**		
1	1	1	0	bits 12-15	1	0	bits 6-11	1	0	bits 0-5

Notes on encoding rules

The first bits of each byte indicate the role of the byte. A zero bit terminates this role information. Thus possible byte values are:

Table -5: UTF-8 Encoding Rules

Bits	**Byte value**	Role
0???????	000-127	Single byte encoding of a character
10??????	128-191	Continuation of a multi-byte encoding
110?????	192-223	First byte of a two byte character encoding
1110????	224-239	First byte of a three byte character encoding
1111???	240-255	Invalid

Example encoding

Table -5: UTF-8 Encoding Example

Character	**S**	**C**	**T**	®		③		
Unicode	0053	0043	0054	00AE		2462		
Bytes	01010011	01000011	01010100	11000010	10101110	11101111	10111111	10111111

Appendix D: Concept Definition Illustrations

A concept definition is A set of one or more <a href="https://confluence.ihtsdotools.org/display/DOCGLOSS/axiom" title="Glossary link: axioms" rel="nofollow" class="conf-macro output-inline" data-hasbody="false" data-macro-name="gloss">axioms</a> that partially or sufficiently specify the meaning of a <a href="https://confluence.ihtsdotools.org/display/DOCGLOSS/SNOMED+CT+concept" title="Glossary link: SNOMED CT concept" rel="nofollow" class="conf-macro output-inline" data-hasbody="false" data-macro-name="gloss">SNOMED CT concept</a>.</p> <div class="NoPrint" style="margin-top:20px;"> a set of one or more axioms that partially or sufficiently specify the meaning of a SNOMED CT concept.

Notes

- The axioms that specify a *concept definition* are represented in release files as SNOMED CT relationships or as OWL axioms that conform to the OWL Functional Syntax.

This appendix illustrates some of the features of concept definitions outlined in section 2.3 Concept Definitions.

⚠ The information in this section is particularly relevant to those wishing to understand the changes being made to enhance the representation of stated concept definitions between July 2018 and 2019. The transitional period for these changes begins with the July 31 release of the SNOMED CT International Edition and is scheduled to be completed during 2019.

D.1 Stated and Inferred Definitions - Examples

The appendix contains an extended version of 2.3.1 Stated and Inferred Concept Definitions supported by more detailed examples.

Stated View of Concept Definitions

SNOMED CT concepts are defined by assertions made by SNOMED CT authors. The concept definitions asserted by SNOMED CT authors are known as the stated view.

The stated view is a representation of concept definitions consisting only of assertions made or revised by SNOMED CT authors.

Notes

- In contrast to the inferred view, the *stated view* does not include inferences generated by applying a description logic classifier.

Description Logic Classification

A description logic classifier can apply logical rules to the stated view to create inferences. The end result of this process is an inferred view of concept definitions.

Inferred Views of Concept Definitions

The inferred view is a representation of concept definitions that is logically derived by applying a description logic classifier to the stated view.

Notes

- Different *inferred views* can be derived from the same stated view by applying different rules that selectively exclude some types of assertions.
- Different *inferred views* may be semantically equivalent to one another provided that assertions are only excluded if they are redundant (i.e. can be *inferred* from assertions that are included). However, in some cases, an *inferred view* may not completely represent the concept definition but may serve a specific purpose.

Illustration of the Effect of Classification

Table D.1-3 shows the stated view of the definitions of 710785000 |Laparoscopic repair of hernia|. Compare this with the inferred view of the same concept in Table D.1-3 and you can see that the single supertype concept 71388002 |Procedure| been replaced by four supertype concepts in the inferred view.

The classifier has compared the definition of 710785000 |Laparoscopic repair of hernia| with the concept definitions shown in Table D.1-3 and determined that 710785000 |Laparoscopic repair of hernia| is a subtype of all these concepts.

In fact, the classifier will also have found several other supertypes but the inferred view distributed in the relationship file only includes proximal supertypes (parents). Other supertype ancestors are excluded from the file because they are redundant[1] .

Table D.1-1: Stated view of the definition of |Laparoscopic repair of hernia|

Concept	Stated View of Concept Definition
710785000 |Laparoscopic repair of hernia|	=== 71388002 |Procedure| : { 363700003 |Direct morphology| = 414402003 |Hernial opening (morphologic abnormality)| , 425391005 |Using access device| = 86174004 |Laparoscope, device| , 260686004 |Method| = 257903006 |Repair - action| }

Table D.1-2: Inferred view of the definition of the concept |Laparoscopic repair of hernia|

Concept	Inferred View of Concept Definition
710785000 |Laparoscopic repair of hernia|	=== 363321000 |Surgical repair procedure by device| + 50465008 |Hernia repair| + 51316009 |Laparoscopic procedure| + 264274002 |Endoscopic operation| : { 363700003 |Direct morphology| = 414402003 |Hernial opening (morphologic abnormality)| , 425391005 |Using access device| = 86174004 |Laparoscope, device| , 260686004 |Method| = 257903006 |Repair - action| }

Table D.1-3: Stated views of the four supertype concepts in the inferred |Laparoscopic repair of hernia|

Concept	Stated View of Concept Definition
363321000 |Surgical repair procedure by device|	=== 4365001 |Surgical repair| : { 405815000 |Procedure device| = 49062001 |Device| , 260686004 |Method| = 257903006 |Repair - action| }
50465008 |Hernia repair|	=== 4365001 |Surgical repair| : { 363700003 |Direct morphology| = 414402003 |Hernial opening (morphologic abnormality)| , 260686004 |Method| = 257903006 |Repair - action| }

4365001 |Surgical repair| A supertype in the two definitions above	=== 128927009 |Procedure by method| : 260686004 |Method| = 257903006 |Repair - action|
51316009 |Laparoscopic procedure|	=== 363687006 |Endoscopic procedure| : 425391005 |Using access device| = 86174004 |Laparoscope, device|
363687006 |Endoscopic procedure| A supertype in the definition above	=== 71388002 |Procedure| : { 425391005 |Using access device| = 37270008 |Endoscope, device| , 260686004 |Method| = 129284003 |Surgical action| }
264274002 |Endoscopic operation|	=== 71388002 |Procedure| : { 425391005 |Using access device| = 37270008 |Endoscope, device| , 260686004 |Method| = 129284003 |Surgical action| }

Supertype ancestor relationships are not included in the inferred view distributed in the relationship file because they do not contribute directly to the concept definition and can be reached transitively.

D.2 Necessary and Sufficient - Examples

The appendix contains an extended version of 2.3.2 Necessary Conditions and Sufficient Definitions supported by more detailed examples.

Assertions

The stated view of concept definition consists of one or more assertions made by SNOMED CT authors.

Necessary Conditions

Each time an assertion is made about a concept, an author must decide if that assertion is a necessary condition. If the assertion is always true for that concept and its subtypes, it is a necessary condition.

- This implies that for all instances of that concept or its subtypes, the assertion must be true, even if it has not been explicitly stated.

A necessary condition is defined as a characteristic that is always true of a concept.

Example

- If you have a 71620000 |fracture of femur|, the morphological abnormality 72704001 |fracture| must be present. Therefore, 116676008 |morphology| = 72704001 |fracture| is a *necessary condition* of 71620000 |fracture of femur|.

Sufficient Definitions

For each concept an author must decide if there are one or more sets of assertions that form a sufficient definition of that concept. A set of assertions is a sufficient definition if it distinguishes a concept and its subtypes from other concepts.

- This implies that if all assertions in the set are true for a concept, it must be an instance of the defined concept or a subtype of that concept.

A sufficient definition is a set of characteristics which distinguish a concept and its subtypes from all other concepts.

Notes

- Any concept that matches the *sufficient definition* is equivalent to or a subtype of the defined concept.
- A concept may have more than one *sufficient definition*. In that case any concept that matches at least one of these *sufficient definitions* is equivalent to or a subtype of the defined concept.

Examples

- The following set of assertions is a sufficient definition for 74400008 |appendicitis (disorder)| because any concept for which this set of assertions is true must either be the disorder *appendicitis* or a subtype of *appendicitis*.

```
18526009 |disorder of appendix| +
    302168000 |inflammation of large intestine| :
    116676008 |associated morphology| = 23583003 |inflammation| ,
    363698007 |finding site| = 66754008 |appendix structure|
```

- Both the following sets of assertions are sufficient definitions for the concept 8801005 |Secondary diabetes mellitus (disorder)|:

```
73211009 |Diabetes mellitus| : 246075003 |Causative agent| = 105590001 |Substance|
```

```
73211009 |Diabetes mellitus| : 42752001 |Due to| = 64572001 |Disease|
```

- While each of the assertions 246075003 |Causative agent| = 105590001 |Substance| and 42752001 |Due to| = 64572001 |Disease| form part of a sufficient definition, neither of these assertions are necessary conditions because *only one* of them needs to be true. This illustrates that an assertion that is part of a sufficient definition need not be a necessary condition.

Concepts with no Sufficient Definitions

A concept that has no sufficient definitions is a primitive concept.

Because primitive concepts have no sufficient definitions it is not possible for a description logic classifier to determine if other concepts are subtypes of this concept. Similarly, it is not possible to automatically determine whether an expression is a subtype of a primitive concept. Therefore, only concepts or expressions that explicitly state they are subtypes of primitive concepts will be treated as subtypes when applying expression constraints or undertaking analysis.

However, note that this does not prevent a primitive concept being classified as a subtype of a sufficiently defined concept.

Concepts with a Sufficient Definition

A concept that has at least one sufficient definition is a sufficiently defined concept.

A description logic classifier can determine whether the stated definitions of other concepts meet at least one of the sufficient definitions and if so will classify these concepts as its subtypes. Similarly, it is possible to determine whether an expression is equivalent to or a subtype of a sufficiently defined concept. Therefore, where expression constraints or queries refer to sufficiently defined concepts the results will include the inferred subtypes of these concepts.

Sufficiently Defined Concepts with Necessary Conditions

If a sufficiently defined concept has one or more additional necessary conditions then any concept or expression that satisfies one of its sufficient definitions will also inherit any necessary conditions.

For example one sufficient definition of 397825006 |Gastric ulcer (disorder)| is an ulcer in a stomach structure:

```
=== 64572001 |disease| :{ 116676008 |associated morphology| = 56208002 |ulcer| ,
    363698007 |finding site| = 69695003 |stomach structure| }
```

However, another definition could be created with a more specific site gastric mucosa:

```
=== 64572001 |disease| :{ 116676008 |associated morphology| = 56208002 |ulcer| ,
    363698007 |finding site| = 78653002 |gastric mucosa| }
```

In both cases these definition are equivalent to 397825006 |Gastric ulcer (disorder)|. The more general definition is flexible when it comes to allowing refinement to a specific location of the ulcer within the stomach, which is actually useful information. It also avoids requiring an expression to refer specifically to the mucosa (stomach lining), which is where all gastric ulcers occur.
For example, an expression including the specific location could look like this

```
=== 64572001 |disease| :{ 116676008 |associated morphology| = 56208002 |ulcer| ,
    363698007 |finding site| = 127869006 |Anterior wall of fundus of stomach| }
```

This satisfies the sufficient definition because the finding site is a subtype of stomach structure. This will therefore classify as a type of 397825006 |Gastric ulcer (disorder)| located in the anterior wall of the gastric fundus. The problem is that a query for disorders of the gastric mucosa will not find this expression. << 64572001 |disease| : 363698007 |finding site| = 78653002 |gastric mucosa| However, adding the definition that refers to the gastric mucosa as an additional necessary condition can solve this problem. The expression satisfies the sufficient definition implying this is a type of 397825006 |Gastric ulcer (disorder)|. The fact that it is a type of gastric ulcer causes it to inherit 363698007 |finding site| = 78653002 |gastric mucosa| so it will now be included in the query for disease in the gastric mucosa.

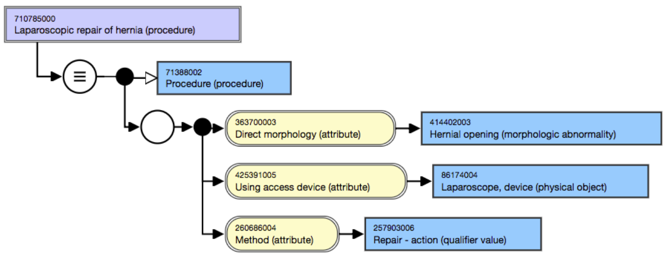
A Definition that is Both Necessary and Sufficient

The definition shown in Table D.2-3 provides an example of a simple case.

- The === symbol indicates that the concept definition is equivalent to the concept.
    - This means that each of the assertions in the definition is **necessarily** true for all instance of the concept 710785000 |Laparoscopic repair of hernia|.
    - It also means that this definition is **sufficient**, because if all the assertions are true, this implies this is either the concept or a subtype of the concept.

Table D.2-1: Stated view of the definition of |Laparoscopic repair of hernia|

Concept	Stated View of Concept Definition
710785000 |Laparoscopic repair of hernia|	=== 71388002 |Procedure| : { 363700003 |Direct morphology| = 414402003 |Hernial opening (morphologic abnormality)| , 425391005 |Using access device| = 86174004 |Laparoscope, device| , 260686004 |Method| = 257903006 |Repair - action| }

710785000
Laparoscopic repair of hernia (procedure)
≡
71388002
Procedure (procedure)
363700003
Direct morphology (attribute)
414402003
Hernial opening (morphologic abnormality)
425391005
Using access device (attribute)
86174004
Laparoscope, device (physical object)
260686004
Method (attribute)
257903006
Repair - action (qualifier value)

The definition shown in Table D.2-3 provides an example of another common case.

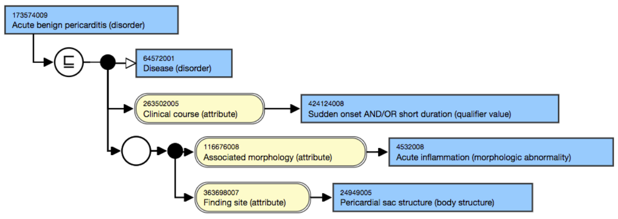
- The === symbol indicates that the concept definition is equivalent to the concept.
    - This means that each of the assertions in the definition **necessarily** true for all instance of the concept 710785000 |Laparoscopic repair of hernia|.
    - It also means that this definition is **sufficient**, because if all the assertions are true, this implies this is either the concept or a subtype of the concept.

A Definition that is Necessary but Not Sufficient

The definition shown in Table D.2-3 provides an example of another simple case.

- The <<< symbol indicates that the concept is a subtype of the concept definition.
    - This means that each of the assertions in the definition is **necessarily** true for all instance of the concept 173574009 |Acute benign pericarditis (disorder)|.
    - However, this definition is **not sufficient**, because it is represent a more general meaning. Put another way, it does not capture one or more distinguishing features or the 173574009 |Acute benign pericarditis (disorder)|. This means that even if all the assertions are true, it may or may not be this concept or one of its subtypes.

Table D.2-2: Stated view of the definition of |Acute benign pericarditis|

Concept	Stated View of Concept Definition
173574009 |Acute benign pericarditis (disorder)|	<<< 64572001 |Disease| : 263502005 |Clinical course| = 424124008 |Sudden onset AND/OR short duration| { 116676008 |Associated morphology| = 4532008 |Acute inflammation| , 363698007 |Finding site| = 24949005 |Pericardial sac structure| }
	

173574009
Acute benign pericarditis (disorder)
⊑
64572001
Disease (disorder)
263502005
Clinical course (attribute)
424124008
Sudden onset AND/OR short duration (qualifier value)
116676008
Associated morphology (attribute)
4532008
Acute inflammation (morphologic abnormality)
363698007
Finding site (attribute)
24949005
Pericardial sac structure (body structure)

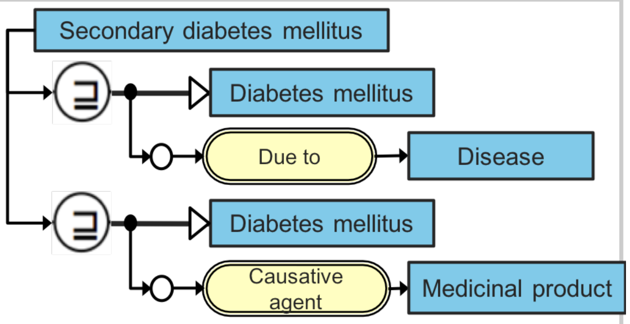
A Definition that is Sufficient with Assertions that are Not Necessarily True

This example is illustrates a type of definition that was not supported prior to the enhancement to SNOMED CT support for advanced description logic. Therefore, the definition shown is for illustration only and will not be found in current releases of SNOMED.

The definition shown in Table D.2-3 provides an example of a more complex case.

- The >>> symbol indicates that the concept definition represents a subtype of the concept definition.
  - This means that each of the assertions in the definition is **necessarily** true for all instance of the concept 173574009 |Acute benign pericarditis (disorder)|.
  - However, this definition is **not sufficient**, because it is represent a more general meaning. Put another way, it does not capture one or more distinguishing features or the 173574009 |Acute benign pericarditis (disorder)|. This means that even if all the assertions are true, it may or may not be this concept or one of its subtypes.

Table D.2-2: Possible stated view of the definition of |Secondary diabetes mellitus|

Concept	Stated View of Concept Definition
8801005 |Secondary diabetes mellitus (disorder)|	>>> 73211009 |Diabetes mellitus| : 246075003 |Causative agent| = 105590001 |Substance| OR >>> 73211009 |Diabetes mellitus| : 42752001 |Due to| = 64572001 |Disease|
	 Secondary diabetes mellitus Diabetes mellitus Due to Disease Diabetes mellitus Causative agent Medicinal product

Sufficiently Defined

1. A sufficiently defined concept is a concept with one or more sufficient definitions.

Notes

- A SNOMED CT concept is expressed in a human-readable form by its fully specified name (FSN).
- A *sufficiently defined concept* has at least one sufficient definition that distinguishes it from any concepts or expressions that are neither equivalent to, nor subtypes of, the defined concept.

Examples

- The concept 74400008 |appendicitis (disorder)| is *sufficiently defined* by the following definition because any concept for which these defining relationships are true, is either the disorder *appendicitis* or a subtype of *appendicitis*.

```
74400008 |appendicitis (disorder)|
    === 18526009 |disorder of appendix| :
        116676008 |associated morphology| = 23583003 |inflammation| ,
        363698007 |finding site| = 66754008 |appendix structure|
```

- If a concept has a *sufficient* definition, it is possible to infer whether another concept or a postcoordinated expression is a subtype of, or equivalent to, that concept.

2. Primitive

A primitive concept is a concept without a sufficient definition in the necessary normal form distributed in the relationship.

Notes

- The meaning of a SNOMED CT concept is expressed in a human-readable form by its fully specified name. Each concept also has a formal concept definition that provides a computer-processable representation of the meaning of the concept.
- A *primitive concept* has a concept definition that is not sufficient to computably distinguish it from other concepts.

Example

- The concept 5596004 |atypical appendicitis (disorder)|is *primitive* because the following definition is not sufficient to distinguish *atypical appendicitis* from any other type of *appendicitis:*

```
5596004 |atypical appendicitis (disorder)|
      <<< 116680003 |is a| = 74400008 |appendicitis|
        116676008 |associated morphology| = 23583003 |inflammation|
        363698007 |finding site| = 66754008 |appendix structure|
```

Necessary Conditions

All SNOMED CT defining relationships currently released are necessarily (always) true for the concept defined. Relationships that are necessarily true are also know as necessary conditions.

A necessary condition is defined as a characteristic that is always true of a concept.

Example

- If you have a 71620000 |fracture of femur|, the morphological abnormality 72704001 |fracture| must be present. Therefore, 116676008 |morphology| = 72704001 |fracture| is a *necessary condition* of 71620000 | fracture of femur|.

Sufficient Sets of Conditions

In practice there can be several sufficient definitions for a concept. That is to say several different ways in which a concept could be sufficiently defined by different sets of defining relationships For example:

Gastric ulcer is defined as follows:

```
397825006 |gastric ulcer|
===    116680003 |is a| = 64572001 |disease|
     { 116676008 |associated morphology| = 56208002 |ulcer| ,
      363698007 |finding site| = 69695003 |stomach structure| }
```

This is a *sufficient* definition because any 56208002 |ulcer|in a 69695003 |stomach structure|is by definition a 397825006 |gastric ulcer| .Based on this definition:

Any postcoordinated expression that specified a disease involving an 56208002 |ulcer|with 363698007 |finding site| 69695003 |stomach structure|would be equivalent to or a subtype of 397825006 |gastric ulcer|

However, a query for all disorders involving 78653002 |gastric mucosa|would incorrectly exclude 397825006 |gastric ulcer|as the site is specified as 78653002 |gastric mucosa|which is more specific than 69695003 |stomach structure| .I n reality there is another sufficient set defining relationships

```
397825006 |gastric ulcer|
===    116680003 |is a| = 64572001 |disease|
   { 116676008 |associated morphology| = 56208002 |ulcer| ,
    363698007 |finding site| = 78653002 |gastric mucosa| }
```

but this is not currently represented in SNOMED CT. The reason for this is that currently the profile of description logic used by SNOMED CT does not support representation of multiple sufficient sets.
When multiple sufficient sets are supported, satisfying a single sufficient set enables an inference to be made that all necessary conditions must also be true. For example

- The definition 363698007 |finding site|= 78653002 |gastric mucosa|is a *necessary* condition for 397825006 |gastric ulcer|:
  - This is true because all gastric ulcers necessarily involve the 78653002 |gastric mucosa|
- The definition 116676008 |morphology|= 56208002 |ulcer|and 363698007 |finding site|= 69695003 |stomach structure|is a *sufficient* definition for 397825006 |gastric ulcer|:
  - This is true because any ulcer in a stomach structure is a 397825006 |gastric ulcer|
- Therefore, an assertion that a person has an 56208002 |ulcer|with 363698007 |finding site| 69695003 |stomach|is *sufficient* to imply that they have a 397825006 |gastric ulcer|:
  - Since a gastric ulcer *necessarily* involves the 78653002 |gastric mucosa|it should be possible to deduce that a person with an "ulcer" with finding site 69695003 |stomach|has a disorder of with a site 78653002 |gastric mucosa|

However, as the current profile does not enable recognition of multiple sufficient sets, the general rule is to represent the most general sufficient set as this gives the greatest coverage for subsumption testing. This approach is taken because including more defining relationships, without distinguishing them from the sufficient set means some logically equivalent expressions will not compute as equivalent to or subsumed by the defined concept. This occurs in any cases where the expression does not include one of the attributes in the definition - even if it was not part of the logically sufficient set.