INTERNATIONAL HEALTH TERMINOLOGY
STANDARDS DEVELOPMENT ORGANISATION

SNOMED CT® Release Format 1 (RF1) Guide

July 2014 International Release
(US English)

This document may have been updated since you received it.
The latest versions of IHTSDO documents are available online.
Latest PDF version: www.snomed.org/rf1.pdf
Latest web browsable version: www.snomed.org/rf1
Directory of available documents: www.snomed.org/doc

Contents

Chapter 1

1 Preface

The purpose of the Release Format 1 Guide is to meet the needs of people who require an authoritative point of technical reference to *Release Format 1* (RF1). RF1 was used to distribute the *International Release* of *SNOMED CT* until July 2011. Since January 2012 the file formats described in this document have been replaced by the enhanced *Release Format 2* (RF2). However, for a transitional period, RF1 distribution files are still available, either for download or by conversion of RF2 *release files* using the *IHTSDO* RF2 to RF1 Conversion Tool.

Note: While this guide covers the file format specifications and specific RF1 aspects of *Terminology services*, other useful *Release Format* independent information can be found in the *SNOMED CT Technical Implementation Guide*.

Important: RF1 specifications and guidance are no longer maintained or updated. In addition, RF1 files do not support the full range of versioning and extensibility features of RF2. Therefore, this guide is provided only for use by those who need to continue to support RF1 during a transitional period. Anyone engaged in new development of *SNOMED CT* software is strongly recommended to implement support for RF2 as soon as possible. Refer to the *SNOMED CT Technical Implementation Guide* for *RF2 Specifications* and *Guidance on Terminology Services* required to fully support RF2 functionality.

1.1 Document Properties

Table 1:

Title:	*SNOMED CT Release Format 1* Guide
Date	2012-07-31
Version	2012-07-31 *International Release*
Creating Author:	David Markwell
Subject:	Retained legacy specification and guidance specific to *SNOMED CT Release Format 1*.

1.2 Amendment History

Table 2: Amendment History

Version	Date	Editor	Comments
1.00	2002 02 31	David Markwell	First draft of TIG provided with first *SNOMED CT release* accompanied by technical documents on *Release Format*.
12.00	2012 01 31	David Markwell	Last version of the TIG that contained this *Release Format 1* specific documentation.
13.0	2012 07 31	David Markwell	*Release Format 1* specific material removed from the Technical Implementation Guide but retained for backward compatibility in this separate document.

1.3 Status

This guide contains parts and sections which differ in terms of the authority and status of their content. Each section of the guide is marked to indicate its publication type and status using the symbols shown in *Table 1* and *Table 2*.

Table 3: Document Types

Type Name and *Description*	Draft	Review	Current
Standard A document or other resource that is intended to be authoritative. This includes specifications of *SNOMED CT* content and *release files*. Normative requirements for particular functions are also standards.			
Guidance A document or other resource that is intended to provide advice or suggest possible approaches to particular requirement or subject area.			

Table 4: Document Status

Status Name and *Description*	Standard	Guidance
Current *Indicates that the document or resource is considered to be up-to-date and complete for the current release of SNOMED CT (indicated by an explicitly stated version date or by the publication date).*	S	G
Review *Indicates that the document or resource has been released for review and comments from SNOMED CT users and other stakeholders. It is intended to be complete but has not been formally approved as a final version.*	S	G
Draft *Indicates that the document or resource is a draft version. It may be incomplete and has not been approved in a final version.*	S	G

This edition of the document is configured to use US English .

The PDF version of this draft is formatted to be printed on US Letter paper.

Note: This is one of a several large documents that are regularly revised by the *IHTSDO*. Therefore, for the sake of the environment, please think carefully before deciding to print the entire document.

1.4 Referencing and Commenting

This document contains a way to reference topics in a way that is not dependent on changes to the structure of the document as new versions are released including additional topics. These references are web addresses that will point to the latest version of and topic in the document.

If you are using the PDF version of the document there are three icons to the right of each title which provide useful information and relevant links.

- The S icon indicates the status of the topic (see *Status*).
- The WWW icon provides a link to the web address to access and reference this topic online. Please use this reference to identify or share references to the topic as section and page numbers change between versions.
- The icon links directly to a page where you can submit comments or report errors about this topic. The comment tracker is an online resource that requires you to login to an IHTSDO CollabNet account. If you do not have an account, there is an option to create an account available on the login page.

If you are using the online web version of this document then there is a single bookmark icon which, when clicked, opens a small form with an easy copy and paste option for access to the topic reference and button to click to take you direct to the comment tracker.

1.5 Additional information

Further information about *SNOMED CT* is available by contacting *IHTSDO*:

IHTSDO Contact Details:

Web:

- *www.ihtsdo.org*

Email:

- *support@ihtsdo.org*

Address:

- IHTSDO
- Gammeltorv 4, 1.
- 1457 Copenhagen K
- Denmark
-
- Tel: +45 3644 8736
- Fax: +45 4444 8736

1.6 Inventory of Documentation

The following *SNOMED CT* documentation is made available to accompany the *International Release* of *SNOMED CT* from the International Health Terminology Standards Development Organization (*IHTSDO*). In the following listing hyperlinks are provided which will be maintained to point to the latest version of each of these documents.

- A list of documents, including a wider range of versions, is available from: *www.ihtsdo.org/doc*.

***SNOMED CT* Technical Implementation Guide (TIG)**

- On line HTML version: *www.ihtsdo.org/tig*
- PDF version US English Letter page size: *www.ihtsdo.org/tig.pdf*
- PDF version UK English A4 page size: *www.ihtsdo.org/tig_gb.pdf*

The TIG is intended for *SNOMED CT* implementers, such as software designers. The TIG assumes information technology and software development experience. Clinical knowledge is not required, although some background is helpful to understand the application context and needs.

The TIG contains guidelines and advice about the design of applications using *SNOMED CT*, and covers topics such as *Terminology services*, entering and storing information, and migration of legacy information.

***SNOMED CT* Editorial Guide**

- On line HTML version: *www.ihtsdo.org/eg*
- PDF version US English Letter page size: *www.ihtsdo.org/eg.pdf*
- PDF version UK English A4 page size: *www.ihtsdo.org/eg_gb.pdf*

The Editorial Guide is intended for clinical personnel, business directors, software product managers, and project leaders; information technology experience, though not necessary, can be helpful.

The Editorial Guide is intended to explain *SNOMED CT*'s capabilities and uses from a content perspective. It explains the content and *concept model*, and the principles used to edit the terminology.

***SNOMED CT* User Guide**

- On line HTML version: *www.ihtsdo.org/ug*
- PDF version US English Letter page size: *www.ihtsdo.org/ug.pdf*
- PDF version UK English A4 page size: *www.ihtsdo.org/ug_gb.pdf*

The User Guide provides a less detailed introduction to the topics covered in the Technical Implementation and Editorial Guides.

***IHTSDO* Glossary (DRAFT)**

- On line HTML version: *www.ihtsdo.org/glossary*
- PDF version US English Letter page size: *www.ihtsdo.org/glossary.pdf*
- PDF version UK English A4 page size: *www.ihtsdo.org/glossary_gb.pdf*

The Glossary is a general resource used to support all the other documents in this inventory.

***SNOMED CT Release Format 1* Guide**

- On line HTML version: *www.ihtsdo.org/rf1*
- PDF version US English Letter page size: *www.ihtsdo.org/rf1.pdf*
- PDF version UK English A4 page size: *www.ihtsdo.org/rf1_gb.pdf*

The RF1 Guide provides technical information relevant to those using the original *SNOMED CT Release Format*. Although this format was replaced by RF2 in January 2012, the old format is being maintained for a transitional period.

***SNOMED CT* Non-Human *Refset* Guide**

- PDF version US English Letter page size: *www.ihtsdo.org/guide/non_human_rs.pdf*

A guide to use of the "Non-Human" Simple *Reference Set* that contains *concepts* and terms that are only used in veterinary medicine.

***SNOMED CT* Developer Toolkit Guide**

- PDF version US English Letter page size: *www.ihtsdo.org/guide/toolkit.pdf*

A guide to use of value-added files and scripts that are provided as a toolkit available as part of the *SNOMED CT International Release*.

Additional Documentation: The following materials previously published in separate documents are now integrated as part of the Technical Implementation Guide.

- *Technical Reference Guide*
- *Namespace Identifier Guide*
- *Namespace Identifier Registry*
- *File Naming Convention*
- *RF2 Data Structures Specification*
- *RF2 Reference Set Specifications*
- *RF2 Update Guide*
- *Stated Relationships Guide*
- *Canonical Table Guide* (previously included in RF1)

1.7 Where is the Glossary?

Some versions of documents may contain a glossary section. However, we are also developing a separate *IHTSDO* Glossary document which is currently available in a draft form. The intention is to move toward using this single common resource make it easier to ensure consistency across the *IHTSDO* community.

The current version of the *IHTSDO* Glossary is available as follows:

- On line HTML version: *www.ihtsdo.org/glossary*
- PDF version US English Letter page size: *www.ihtsdo.org/glossary.pdf*
- PDF version UK English A4 page size: *www.ihtsdo.org/glossary_gb.pdf*
- You can create links that query the glossary use the following web address pattern "www.ihtsdo.org/define/*word or phrase*". The following examples can be tested and you can include these types of reference in your documents to make it easy to refer to the *IHTSDO* glossary definitions:
  - *www.ihtsdo.org/define/ihtsdo*
  - *www.ihtsdo.org/define/snomed ct*
  - *www.ihtsdo.org/define/affiliate licence*

1.8 Copyright Notice

Chapter 2

2 Structure and Content Guide

This part of the guide covers the features of *SNOMED CT* that need to be understood by those implementing *SNOMED CT* in software applications. These features include the components, *derivatives* and supporting materials that are distributed as part of each *SNOMED CT Release*. In addition, the guide addresses the ways in which these components may be referenced to represent instances of clinical information in clinical records and other types of instance data.

2.1 SNOMED CT Technical Overview (RF1)

This section provides an overview of the *components* and *derivatives* that form part of a *SNOMED CT release* as well as several other topics that relate to the use of *SNOMED CT* to represent instances of clinical information.

These topics are explored in more depth by other sections in this part of the guide:

- *Logical Abstract Models*;
- *Representational Forms*.

More detailed information about technical design and content is provided in other parts of the guide:

- *Release File Specifications*;
- *Concept Model Guide*.

2.1.1 Components

This section summarizes the essential *components* of *SNOMED CT* (*concepts*, *descriptions* and *relationships*). A *SNOMED CT enabled implementation* must be able to process and make appropriate use of these *components*, which are distributed as a set of *Release Files*.

2.1.1.1 Concepts

A *SNOMED CT Concept* is a clinical idea to which a unique *SNOMED CT identifier* has been assigned.

Each *Concept* is associated with:

- A unique human-readable *Fully Specified Name* (FSN), which specifies the meaning represented by the *Concept*.
- A set of other *Descriptions*, each of which represents the same *Concept* using a different human-readable *term*. These *Descriptions* support alternative representations such as *synonyms* and translations into different *languages*.
- A set of *Relationships* to other *Concepts* which provide a logical definition of the *Concept* that can be processed by a computer.

2.1.1.1.1 Concept Identifiers

Each *SNOMED CT Concept* has a permanent unique numeric *Identifier* which is known as the *Concept Identifier*.

The sequence of digits in a *Concept Identifier* does not convey any information about the meaning or nature of the *Concept* [1]. The meaning of *Concept* is represented in human-readable forms by *Descriptions* and in a computer processable form by *Relationships* with other *Concepts*.

The advantages of meaningless *Identifiers* include:

- *Identifier* permanence without undermining interpretation:
  - In contrast, to maintain consistency, a meaningful code may need to change to reflect revised understanding of the nature of a disorder. .
- Enabling multiple aspects of meaning to be represented in the same way:
  - A meaningful code can only represent part of meaning of a complex *concept*. For example, |staphylococcal pneumonia| is an |infection|, a |respiratory disorder| and a |disorder| caused by |staphylococcus| but only one of these aspects can be represented by a code based *hierarchy*. Thus in the 'J' in the *ICD-10* code 'J152: Pneumonia due to staphylococcus' represents that fact that this is a respiratory disorder but does not represent the fact that it is an infection (codes starting with 'A') or that it is due to staphylococcus ('A490: Staphylococcal infection, unspecified').
- No artificial limitation on *concept* granularity:
  - Typical approaches to meaningful coding impose limits on both the number of levels of specificity (i.e. the length of the code) and the number of options at each level (i.e. the number of different symbols that can be used in each character position).

2.1.1.1.2 Concept granularity

The meaning represented by a *Concept* can be general (for example | procedure |), specific (for example | excisional biopsy of lymph node |) or somewhere in between (for example | biopsy of lymph node |).

- More specific *Concepts*:
  - Have finer granularity (more granular);
  - Represent clinical detail.
- More general *Concepts*:
  - Have coarser granularity (less granular);
  - Represent less clinical detail;
  - Aggregate similar *Concepts*.

Support for multiple levels of granularity allows *SNOMED CT* to be used to represent clinical data at a level of detail that is appropriate to a range of different uses.

Concepts with different levels of granularity are linked to one another by | is a | *relationships*. This enables appropriate aggregation of specific information within less detailed categories.

[1] The use of meaningless identifiers differs from the approach taken by some other coding systems and classifications. For example, the first character of an ICD-10 code indicates the general classification that it falls within.

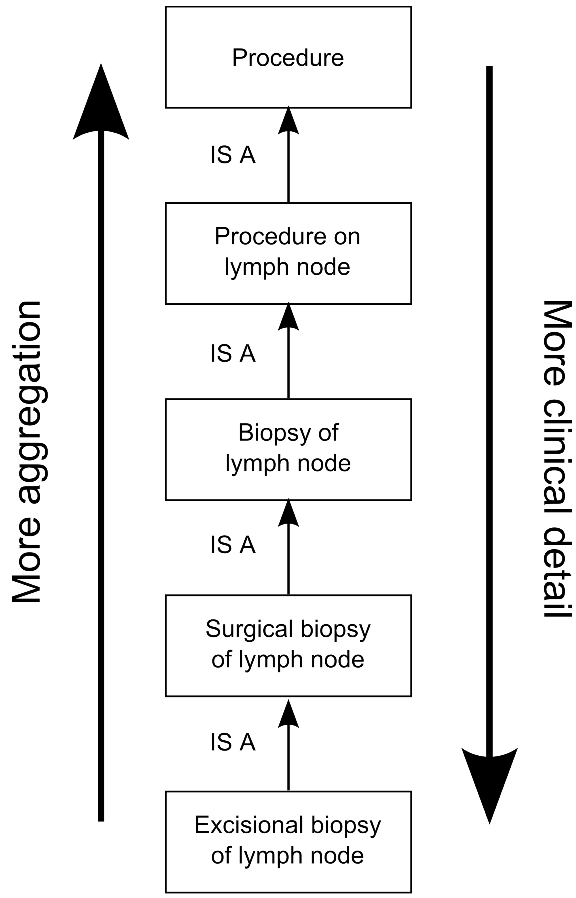


Figure 1: Multiple levels of granularity

2.1.1.2 Descriptions

A *Description* associates a human-readable *term* with a *Concept* that it describes.

A *Concept* is associated with several *Descriptions*. Each of these represents either a *Preferred Term*, *Synonym*, or *Fully Specified Name* for the *Concept* in a particular *language* or *dialect*.

A *Description* may be a *Preferred Term* in one *dialect* and a *synonym* in another *dialect*. This is indicated by references to the *Description* from an appropriate Language *Reference Set*.

Each *Description* is identified by a unique *Description Identifier* and is distributed as a row in the *Descriptions Table*.

2.1.1.2.1 Fully Specified Name

Each *concept* has one *Fully Specified Name* (FSN) intended to provide an unambiguous way to name a *concept*. The purpose of the FSN is to uniquely describe a *concept* and clarify its meaning. The FSN is not a commonly used term or natural phrase and would not be expected to appear in the human-readable representation of a clinical record.

Note: The term in each FSN is unique across the entire active content of a *SNOMED CT release*.

Each FSN term ends with a “semantic tag” in parentheses. The semantic tag indicates the semantic category to which the *concept* belongs (e.g. clinical finding, disorder, procedure, organism, person, etc.). The “semantic tag” helps to disambiguate the different *concept* which may be referred to by the same commonly used word or phrase.

Example: | Hematoma (morphologic abnormality) | is the FSN of the *concept* that represents the “hematoma” that a pathologist sees at the tissue level. In contrast, | Hematoma (disorder) | is the FSN of the *concept* that represents the clinical diagnosis that a clinician makes when they decide that a person has a "hematoma".

2.1.1.2.2 Preferred Term

Each *concept* has one *Preferred Term* in a given language *dialect*. The *Preferred Term* is a common word or phrase used by clinicians to name that *concept*.

Example: the *concept* 54987000 | repair of common bile duct (procedure) | has the *Preferred Term* | choledochoplasty | to represent a common name clinicians use to describe the procedure.

Note: Unlike the *Fully Specified Name* (FSN) the *Preferred Terms* need not be unique. Occasionally, the *Preferred Term* for one *concept* may also be a *Synonym* or the *Preferred Term* for a different *concept*. Interpretation in these cases will depend on context of use.

Example:

- | Cold sensation quality (qualifier value) | has a *preferred term* of "Cold";
- | Common cold (disorder) | also has a *synonym* of "Cold".

In both cases, "cold" represents a common clinical phrase used to capture the meaning of the *concept*.

Note: Selection of one term over another as "preferred" in a given language *dialect* depends entirely on whose preferences are being expressed. Different users are likely to have different preferences, and implementers are encouraged to select or create terms that properly represent the *concept* and meet the preferences of users. There is no expectation that the *Preferred Term* distributed with a given language *dialect* will meet all use cases; nor is there anything sacrosanct about the term. The U.S. English *Preferred Term* is not guaranteed to have any special status relative to other terms. Rather, it is merely one term that properly represents the *concept* and can be used as a starting point.

2.1.1.2.3 Synonym

A *synonym* represents a *term*, other than the FSN or *Preferred Term*, that can be used to represent a *concept* in a particular language or *dialect*.

Example: *Synonyms* of the *concept* 22298006 | myocardial infarction (disorder) | in English include:

- | cardiac infarction | (*Description*.id: 37442013);
- | heart attack | (*Description*.id: 37443015);
- | infarction of heart | (*Description*.id: 37441018).

The *Preferred Term* for this *concept* in English is: | myocardial infarction | (*Description*.id: 37436014).

Note: *Synonyms*, like *Preferred Terms*, are not required to be unique.

2.1.1.3 Relationships

A *Relationship* represents an association between two *Concepts*.

Each *Relationship* is identified by a unique *Relationship* Id and is distributed as a row in the *Relationships Table*.

A *Relationship* contains *Identifiers* of two logically associated *Concepts* and the *Identifier* of another *Concept* that indicates the *Relationship Type* by which they are associated.

Table 5: Example: Defining arthritis as a type of joint disorder

Relationship.id	source.id	type.id	destination.id
2227469024	3723001	116680003	399269003
In human readable *terms* ...	| arthritis |	| is a |	| joint disorder |

2.1.1.3.1 Relationships and concept definitions

Each *concept* in *SNOMED CT* is logically defined through its *relationships* to other *concepts*.

Every *active SNOMED CT concept* (except the *SNOMED CT Concept Root concept*) has at least one | is a | *relationship* to a supertype *concept*.

| is a | *relationships* and defining attribute *relationships* are known as the *defining characteristics* of *SNOMED CT concepts*. They are considered defining because they are used to logically represent a *concept* by establishing its *relationships* with other *concepts*. This is accomplished by establishing | Is a | *relationships* with one or more defining *concepts* (called supertypes) and modeling the difference with those supertypes through defining attributes.

Example: | Fracture of tarsal bone (disorder) | is defined as:

- | is a | *subtype* of | Fracture of foot (disorder) |
- and has | finding site | | Bone structure of tarsus (body structure) | ;
- and has | associated morphology | | Fracture (morphologic abnormality) | .

Note: A *relationship* is assigned only when that *relationship* is always known to be true.

Example: Group A Streptococcus causes most cases of Streptococcal pharyngitis. However, a small percentage of these cases are caused by other species of Streptococcus. Therefore, it would be incorrect to define | Streptococcal sore throat (disorder) | as having | causative agent | | Streptococcus pyogenes (organism) |. Instead it is correctly defined as having the more general | causative agent | | Genus Streptococcus (organism) |.

2.1.1.3.2 IS A Relationships

| is a | *relationships* are also known as “Supertype - *Subtype relationships*” or “Parent - *Child relationships*”. | is a | *relationships* are the basis of *SNOMED CT*'s hierarchies, as illustrated below.

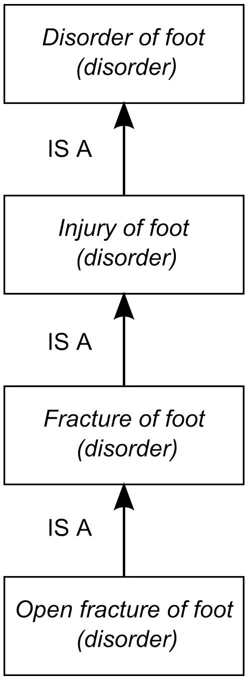


Figure 2: Example IS A hierarchy

A *concept* can have more than one | is a | *relationship* to other *concepts*. In that case, the *concept* will have parent *concepts* in more than one *sub-hierarchy* of a top-level *hierarchy*. *Subtype relationships* can be multi-hierarchical.

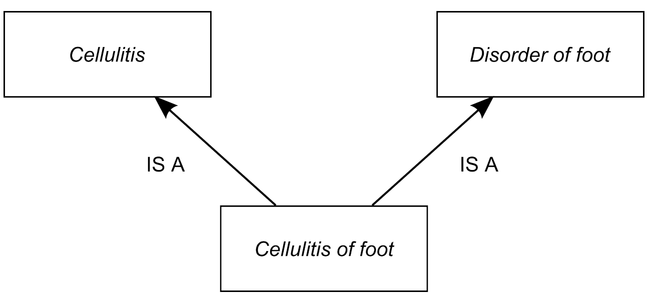


Figure 3: Example IS A Relationships

2.1.1.3.3 Attribute Relationships

An *attribute Relationship* is an association between two *concepts* that specifies a *defining characteristic* of one of the *concepts* (the source of the *Relationship*). Each *Attribute Relationship* has a name (the type of *Relationship*) and a value (the destination of the *Relationship*). For example

The combination of the *attribute Relationships* and | is a | *relationships* associated with a *concept* represent the logical definition of that *concept*. The logical *concept* definition includes one or more supertypes (represented by | is a | *relationships*), and a set of defining *Attributes* that differentiate it from the other *concept* definitions.

Example:

Since pneumonia is a disorder of the lung, the logical definition of the *concept* | Pneumonia (disorder) | in *SNOMED CT* includes the following *Relationship*. The *Attribute* | Finding site | is assigned the value | Lung structure (body structure) |.

- | Finding site | = | Lung structure (body structure) |

The full definitions of the *concepts* | Pneumonia (disorder) |,|Infective pneumonia (disorder)| and |Bacterial pneumonia (disorder)| are shown below. Each line represents a defining *Attribute* with a value.

- | is a | = | pneumonitis |
- , | is a | = | lung consolidation |
- , { | associated morphology | = | inflammation |
- , | associated morphology | = | consolidation |
- , | finding site | = | lung structure | }

Figure 4: Definition of |Pneumonia (disorder)|

- | is a | = | infectious disease of lung |
- , | is a | = | pneumonia |
- , | pathological process | = | infectious process |
- , { | associated morphology | = | inflammation |
- , | associated morphology | = | consolidation |
- , | finding site | = | lung structure | }

Figure 5: Definition of |Infective pneumonia (disorder)|

- | is a | = | bacterial lower respiratory infection |
- , | is a | = | infective pneumonia |
- , | causative agent | = | bacteria |
- , | pathological process | = | infectious process |
- , { | associated morphology | = | inflammation |
- , | associated morphology | = | consolidation |
- , | finding site | = | lung structure | }

Figure 6: Definition of |Bacterial pneumonia (disorder)|

Figure 7 illustrates some of these *Relationships* graphically. | is a | *Relationships* relate a *concept* to more general *concepts* of the same type. In contrast, *Attribute Relationships* (such as | Finding site | and | Causative agent |) relate a *concept* to relevant values in other branches of the *subtype hierarchy*.

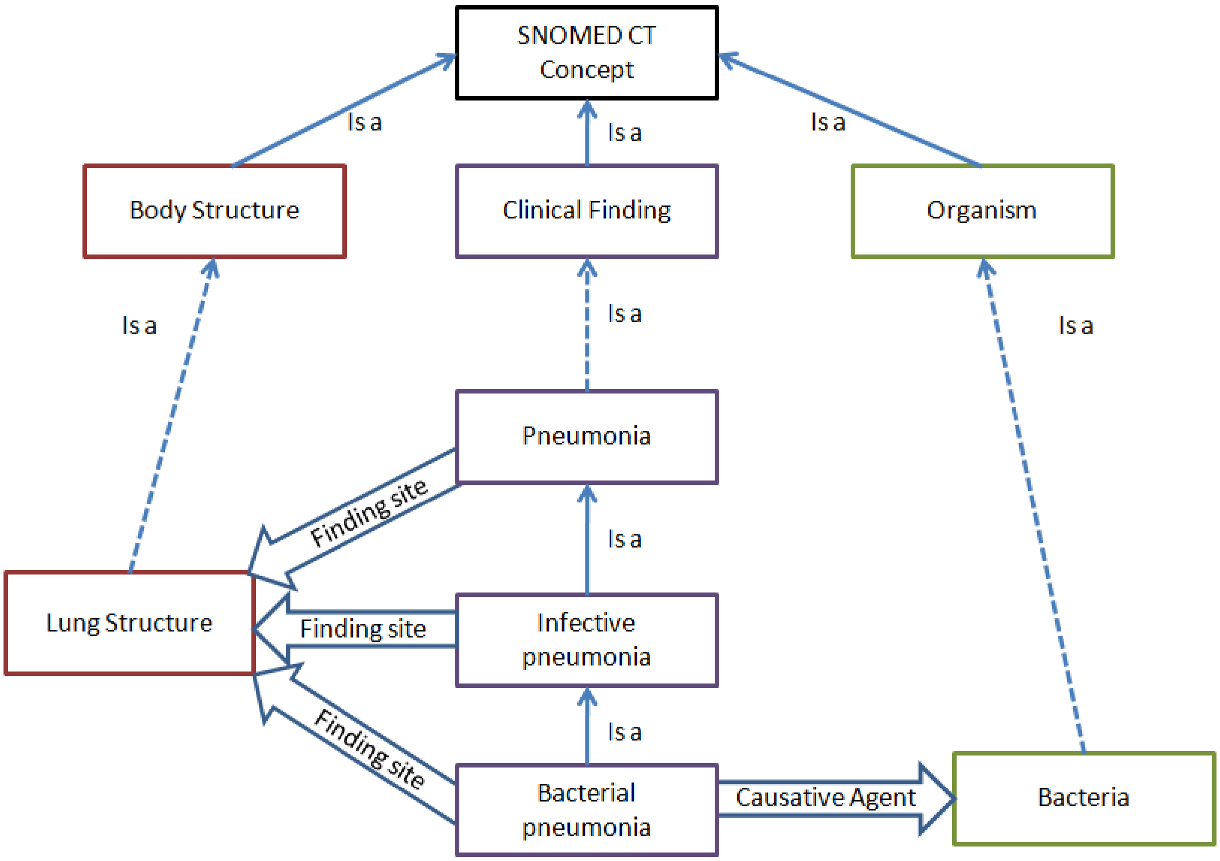


Figure 7: Illustration of Defining Relationships

2.1.1.4 Common Features of Components

This section describes common features of all *SNOMED CT Components* including identification and history management.

2.1.1.4.1 Component features - History

The content of *SNOMED CT* evolves with each release. The types of changes made include new *Concepts*, new *Descriptions*, new *Relationships* between *Concepts*, new *Cross Maps*, and new *Reference Sets*, as well as updates and retirement of any of these *components*. Drivers of these changes include changes in understanding of health and disease processes; introduction of new drugs, investigations, therapies and procedures; and new threats to health, as well as proposals and work provided by *SNOMED CT* users.

Once released, the unique *Identifiers* of *SNOMED CT components* are persistent, and their *Identifiers* are not reused.*Concepts* and *Descriptions* continue to be distributed even when they are no longer recommended for *active* use. This allows a *current* release to be used to interpret data entered using an earlier release.

Since the implementation of *Release Format 2* (RF2), all changes in components are represented in the corresponding files, by adding a new row, with a the same component ID, a new effective time and any necessary change in the component values.

The Component Inactivation *Reference Sets* are used to indicate the *reason* for inactivating a *component*. These *reasons* include errors, duplication of another component and ambiguity of meaning, and the files are used to describe reasons for inactivation of *Concepts*, *Descriptions* and *Relationships*. Some *SNOMED CT Concepts* represent classification *concepts* that have imprecise and potentially changeable meanings. These are marked with the inactivation indicator value [900000000000486000|limited] and were considered *active* until the January 2010 release of *SNOMED CT*. All Limited *concepts* are now considered to be *inactive*.[2]

2.1.1.4.2 Component features - Identifiers

Components within *SNOMED Clinical Terms* 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 *Identifier* to refer unambiguously to a unique component. They also support separate partitions for allocation of *Identifiers* for particular types of component and *namespaces* that distinguish between different issuing organizations .

2.1.2 Derivatives

This section describes *derivatives* that are specified by and distributed as part of *SNOMED CT*. *Derivatives* are artifacts which are either required or useful to support some aspect of *SNOMED CT enabled implementation*. These artifacts are known as *derivatives* because they are derived from *SNOMED CT Components* and either add properties to them or specify sets of related *components*. All *SNOMED CT enabled applications* need to support some *derivatives*.

The set of *derivatives* that need to be supported by an implementation depend on user requirements for particular types of functionality. Important aspects of functionality that require support for relevant *derivatives* include:

- Filtering and prioritizing searches;
- Representing alternative *navigation hierarchies*;
- *Cross Mapping* to and from other coding schemes and classifications.

2.1.2.1 Subsets in RF1

A Subset refers to a set of *Concepts*, *Descriptions*, or *Relationships* that are appropriate to a particular language, *dialect*, country, specialty, organization, user or context.

In its simplest form, the Subset Mechanism is a list of *SNOMED* identifiers (*SCTIDs*). Each *SCTID* refers to one component of *SNOMED CT*, that is, a member of the Subset (called a *Subset Member*). As an analogy, think of *SNOMED CT* as a book. A Subset is like an index entry pointing to a set of pages relevant to a particular topic.

The Subset Mechanism may be used to derive tables that contain only part of *SNOMED CT*. In some cases, these derived tables may also be centrally distributed (e.g. a release table containing only *Descriptions* for a particular *International edition*).

A Subset is a value-added feature of *SNOMED CT*. Subsets provide important information for the use and implementation of *SNOMED CT*. The fact that a *SNOMED CT component* belongs to a particular subset provides information above and beyond the Component itself.

SNOMED CT is a large terminology and subsets can define portions of the terminology for use by specific audiences. For example, a UK *dialect* subset for English may direct the user to *Descriptions* for UK terms rather than all *Descriptions* for English.

Subsets may also be used to specify simplified or shortened hierarchies for browsing, sometimes called "*navigation hierarchies*".

---

[2] Some Concepts derived from classifications such as ICD-10 include the abbreviations NOS (not otherwise specified) or NEC (not elsewhere classified). These are only valid in respect of a particular classification and change in their meaning if additional precisely defined codes are added to that part of the classification. Furthermore, a Concept that is not otherwise specified in ICD-10 may well be more precisely represented by another SNOMED CT Concept and thus from a SNOMED CT perspective "otherwise classified."

Note that it is up to the implementer to determine if a subset is used dynamically or statically, and whether the subset contents are given precedence or used exclusively. Refer to the *SNOMED CT* Technical Implementation Guide for more information.

Note that Subsets are not necessarily mutually exclusive. The contents of Subsets may overlap.

2.1.2.2 Navigation Hierarchies

SNOMED CT subtype Relationships provide a logical semantic *hierarchy*. Often it is possible to view parts of the terminology and select particular *Concepts* by navigating through this *subtype hierarchy*. However, there are many situations in which the pure *subtype hierarchy* does not provide an ideal route for navigating the *hierarchy*.

Navigation links are used to provide an alternative route through parts of the terminology. A *navigation link* can link any two *Concepts* together to identify a useful route for *navigation*. Each of the *navigation links* is directional, linking a navigational parent *Concept* to a more refined navigational *child Concept*. However, unlike the *subtype relationship* the presence or absence of a *navigation link* neither adds to nor subtracts from the definition of either of the *Concepts* that it links.

Some *Concepts* may exist only to provide nodes in a *navigation hierarchy*. These *Concepts* are *subtypes* of |Navigational *Concept* | and play no part in the semantic definitions of any other *Concept*.

2.1.2.2.1 Uses of Navigational Hierarchies

2.1.2.2.1.1 Breaking down a subtype into manageable categories

Some *Concepts* have a large number of *subtype children* that cannot be logically divided into intermediate *subtypes*. At the *user interface* these result in long lists of options, which are difficult to visualize and navigate. Navigational *Concepts* with appropriate navigational links to the *supertype parent* and its *subtype children* provide an intermediate layer without disrupting the semantic definitions.

The | clinical finding | top-level *Concept* has a large number of *subtype children*. Intermediate *navigation Concepts* group some of these together in a convenient way.

Example:

Three *subtypes* related to pregnancy are grouped together under a single natural navigational *Concept*:

- Disorder of pregnancy / labor / delivery / puerperium [*navigation concept*];
- Disorder of pregnancy;
- Disorder of labor / delivery;
- Disorder of puerperium.

2.1.2.2.1.2 Bypassing levels in the subtype hierarchy

Some *Concepts* that are members of the same rational set of choices may be found at different levels in the *subtype hierarchy*. This may occur because some have intervening *subtypes* and some of these intervening *concepts* may not be required for data entry. Addition of new *concepts* in a release may change the *concepts* available at some levels in the *subtype hierarchy*. *Navigation links* can "bypass" levels in the *subtype hierarchy* to represent a rational sets of choices for use in a particular situation.

Example:

While it is semantically correct to nest | common cold | in the following *subtype hierarchy*, a user may reasonably expect to see "common cold" as an immediate navigational *child* of | upper respiratory infection |.

- | upper respiratory infection |
  - | Viral upper respiratory tract infection |
    - | common cold |

2.1.2.2.1.3 Linking related Concepts of different types

Navigational links can also be used to provide access to connected *Concepts* even when they are from different *hierarchy* branches.

Example:

A *navigation links* could associate:

- "hypertension" (the disorder) with | blood pressure | (the observation);
- | cataract | (disorder / finding) with "cataract surgery" (the procedure).

2.1.2.2.1.4 Ordering the display of subtypes

Sibling *Concepts* in a *subtype hierarchy* are not ordered. However, at the *user interface* a particular *order* may be useful to highlight commonly used *Concepts* or to mirror a conventional ordering.

Example:

Vertebrae, cranial nerves, disease stages, etc.

Navigational links are ordered and are used to impose *order*, even when the set of navigational *children* is the same as the set of *subtype children*.

2.1.2.2.1.5 Providing alternative hierarchies

The *subtype hierarchy* is logically defined and there can only be one such *hierarchy*. However, as *navigation* hierarchies have no definitional consequences, it is possible to have different hierarchies for different groups of users with differing needs.

Initial releases of *SNOMED CT* will contain a single set of *navigation links* but those engaged in technical implementation should be aware that in the future there may be separate sets of *navigation links* for use in different environments.

2.1.2.3 Cross Maps

SNOMED CT specifications and content include resources that support *Cross Mapping* to and from other code systems, classifications and terminologies. These resources support simple mapping, where there is a one-to-one *Relationship* between a *SNOMED CT concept* and code in a *target scheme*, and more complex maps where these are required.

More complex mapping requirements supported by the *SNOMED CT Cross Mapping* model include:

- Maps from a single *SNOMED CT concept* to a combination of codes (rather than a single code) in the *target scheme*.
- Maps from a single *SNOMED CT concept* to choice of codes in the *target scheme*. In this case, the resolution of the choices may involve:
  - Manual selection supported by advisory notes.
  - Automated selection based on rules that test other relevant characteristics in the source data (e.g. age and sex of the subject, presence or absence of co-existing conditions, etc).
  - A combination of automated processing with manual confirmation or selection where rules are insufficient to make the necessary decisions.

In *Release Format 2 Cross Maps* are represented using *Reference Sets*. The type of *Reference Set* used varies according to the nature and complexity of the mapping, there is a Simple Map *Reference Set* and a Complex Map *Reference Set*.

2.1.2.4 Search support

The *Developer Toolkit,* which is supplied as part of the *SNOMED CT International Release*, includes several tables that can be used to simplify and support for text searching.

There are two *WordKey Tables*. These tables link each word used in *SNOMED CT* to every:

- *Description* in which it is used;
- *Concept* associated with an *active description* in which the word is used.

There are also two *Dualkey Tables*. These tables link each abbreviated word pair to every:

- *Description* in which that pair of words is used;
- *Concept* in which the combined set of *active descriptions* contains that pair of words.

These tables are provided to assist implementation. However, use of these tables is optional, as developers may generate and use alternative search support resources.

An extended version of the *Developer Toolkit*, provides Java® programs to generate indexes that may be useful to organizations that develop *SNOMED CT Extensions*.

2.1.3 Extensions

SNOMED CT is designed to allow the *International Edition* to be enhanced by adding *Extensions* that meet national or local requirements. *Extensions* are managed by *IHTSDO Members* or *Affiliates* who have been issued with a *Namespace Identifier*, which distinguishes the *Identifiers* of the *Components* they maintain. An *Extension* may contain *Components* of various types (e.g. *Concepts*, *Descriptions*, *Relationships*, and *Derivatives* including *Reference Sets* used for a variety of purposes).

2.1.3.1 Rationale for Extensions

SNOMED CT is a detailed clinical terminology which covers a broad scope. However, some groups of users will need additional *Concepts*, *Descriptions* or *Reference Sets* to support national, local or organizational needs.

This section explains the structures that enable *IHTSDO Members* (*National Release Centers*) and *IHTSDO Affiliates* to add *Concepts*, *Descriptions*, *Relationships* and *Reference Sets* to *complement* the *SNOMED CT International Release*.

The *Extension* mechanism allows *SNOMED CT* to be adapted to address the terminology needs of a country or organization which are not met by the *International Release*. The mechanism provides a structure within which the components of each *Extension* are uniquely identified and attributed to a specific issuing organization. This ensures that, when instance data containing content from different *Extensions* if communicated, the provenance of each referenced *Concepts* is clear and ambiguity is avoided. Since the *International Release* and all *Extensions* share the same common structure, the same application software can be used to enter, store and process information from different extensions. Similarly, *Reference Sets* can be constructed that refer to content from the *International Release* and a variety of *Extensions*.

The common structure also means that, content developed by one organization can where relevant be easily submitted for possible inclusion in a *National Edition* or in the *International Edition*.

Using the *extension* structure can also help organizations transfer responsibility for terminology to the *IHTSDO* or to another organization , subject to the *terms* of the Affiliate License .

- Local content requirements that are likely to have wider applicability should be submitted to a *National Release Center* for consideration.
- National requirements likely to have International value should be submitted to the *IHTSDO* so they can be considered for inclusion in the *International Edition*.

2.1.3.2 Practical uses of Extensions

An *Extension* mechanism offers many advantages to developers, vendors, terminologists, national bodies and users.

Such a mechanism allows:

- **Users** to access the *SNOMED CT International Release* and one or more *Extensions* through a single *user interface*;
- **Developers** to implement *SNOMED CT Extensions* without developing specialized software;
- **Vendors** to develop and sell products to take advantage of both *International Release* content and *Extensions*;
- **Organizations** to develop and share terminology that meet their business needs, without procuring software;
- ***IHTSDO Affiliates*** to develop terminology that can be shared with other organizations and considered for addition to the *International Release* content;
- ***IHTSDO Affiliates*** to use locally-developed terminology without potential overlap with the work of other organizations .

This structure also enables specialized *Concepts* and *Descriptions* within an *Extension* to be related to *Concepts* and *Descriptions* distributed as part of *SNOMED CT*.

- An *Extension Concept* may be:
    - A national or organizational definition of a *concept*, which is more rigorous or specific than that generally applied to the *SNOMED CT Concept*;
    - An experimental procedure that is not established sufficiently to merit the inclusion in the main body of *SNOMED CT* but which may be in a local controlled study.
- *Extension Descriptions* may be colloquial *synonyms* for a *SNOMED CT Concept* or *descriptions* for an *Extension Concept*.
- *Extension Relationships* may be required to allow analysis packages or decision-support protocols to access additional information about a *SNOMED CT Concept* or to describe *relationships* between *Extension Concepts*:
    - Links between local procedures and relevant administrative actions;
    - Links between local procedures and *SNOMED CT* Procedures.
- *Extension Reference sets* may group *SNOMED CT Concepts* in ways that are specific to data entry contexts of a particular application or communication specification.

The *Concepts*, *Descriptions*, *Relationship* and *Reference Sets* that form an *Extension* must be:

- Distinguishable from the main body of *SNOMED CT*, not only in the thesaurus, but also when stored in a patient record, *query* or decision support protocol;
- Distinguishable from other *Extensions*, in the same way as they are distinguishable from the main body of *SNOMED CT*;
- Able to be distributed and processed in the same way as equivalent *components* from the main body of *SNOMED CT* without requiring specific adaptations of *SNOMED-enabled applications*.

The requirements for *Extensions* can be summarized as follows:

- Support for extra terminology *components* including *Concepts*, *Descriptions*, *Relationships* and *Reference Sets*:
    - These extra *components* should behave as though they were *components* of *SNOMED CT* but they should be distinguishable from *components* that are part of the *SNOMED CT International Release*.
- Globally unique identification of any terminology *component* that may be used outside the scope of a limited local environment:
    - The mechanism must allow several organizations to issue mutually exclusive *Identifiers* for *components* of their *Extensions*.
    - To avoid the risk of misinterpretation, this mechanism must be effective in various contexts including:
        - Within the thesaurus;
        - In patient records;

- In queries, decision-support protocols or knowledge bases.

- The mechanism must indicate when *Concepts* have moved, or are expected to move, between an *Extension* and the *International Release*, or from one *Extension* to another.
- A shared understanding of the responsibility of an organization that creates an *Extension* and provides it for the use of other organizations . These responsibilities include:
  - Maintenance of the *Concept*, *Descriptions*, *Relationships*, and *Reference Sets*;
  - Inactivation of these *components* as appropriate (duplication, ambiguous, outdated, etc.);
  - Submission to an *IHTSDO Member*'s *National Release Centre* for consideration as an addition to a *National Edition* or to the *International Release* content.

2.1.4 Instance data

2.1.4.1 Introduction

This section describes the use of *SNOMED CT* to express clinical ideas in patient records, messages, documents, decision support protocols, queries and other artifacts .

Applications need to create, manipulate and consistently interpret standard *SNOMED CT* representations in instance data to support the entry, storage, retrieval and communication of clinical information.

2.1.4.2 Expressions

An *expression* is a structured combination of one or more *concept identifiers* used to express an instance of a clinical idea.

- ***precoordinated expression*: ** An *expression* containing a single *concept identifier* is *precoordinated*. The clinical idea it expresses is represented by the identified *concept*. The defining *relationships* of that *concept* precoordinate its meaning.
- ***postcoordinated expression*: ** *expression* that contains two or more *concept identifiers* is *postcoordinated*. The *concept identifiers* in a *postcoordinated expression* are related to one another in ways that build a more specific clinical idea. The required meaning is expressed by postcoordinating several clinical ideas each of which is represented by an identified *concept*.

Example: A *postcoordinated expression* can indicate the specific site of a finding even when that specific combination of disorder and site is not represented by a single *SNOMED CT Concept*.

2.1.4.3 Terminology Bindings

Terminology binding is one part of the process of specifying *constraints* on the way that information is structured and represented.

Consistent representation is a prerequisite for effective retrieval and reuse of clinical record information. Requirements for reuse are many and varied, ranging from direct support for the care of the individual patient, through to aggregate analysis for research, statistics and audit. The common theme of these requirements is the need to retrieve particular items of information reliably and consistently, irrespective of the environment in which the data was entered and stored.

Since both the information model and *SNOMED CT* contribute to the processable meaning of an entry in a clinical record it is essential to manage the interdependencies between these two components.

Simple requirements can be addressed by specifying a value-set consisting of the permitted coded values that can be used in a particular field. However, effective representation of clinical records requires a rich information model and an expressive terminology.

Models such as *EN13606* and the *HL7* RIM provide the necessary structural flexibility and *SNOMED CT postcoordinated expressions* provide expressivity. An inevitable side-effect of a richer approach to information representation is an increase in the interdependencies and overlaps between the information model and the terminology. In order to specify and validate consistent representation of meaningful clinical records, *constraints* must be applied to both the information model and terminology. These *constraints* must address all the facets of the model and terminology (e.g. including the use of *postcoordination* and the effect of modeled record structures). The *constraints* on information model and terminology components must be integrated, or bound together, in ways that ensure consistency, avoid ambiguity and minimize the number of different ways in which the same meaning may be expressed.

A terminology binding is an instance of a link between a *terminology component* and an *information model artifact .* Therefore, the document considers the representation of the required *terminology components* and the way these are associated with relevant *information model artifacts .*

The *information model artifact* to which a *terminology binding* is applied may be a field of a class in a static model or a collection of *fields* of one or more related classes.

Bound components include:

- Information model artifacts :
    - Coded attributes in an *HL7* Version 3 model, an *EN13606* Archetype or in the proprietary information model of an operational application.
- Terminology components:
    - *Constraints* on *SNOMED CT expressions*.

2.1.4.4 Expression Constraints

SNOMED CT contains hundreds of thousands of *Concepts* and this rich resource is greatly expanded by use of *postcoordinated expressions*. In any given situation the range of *Concepts* or *expressions* that are useful, relevant and meaningful is much more limited. This gives rise to a requirement to represent *constraints* on the content or a particular field in a way that can be interpreted and applied by application software.

The simplest *constraint* requirements can be met by specifying the list of valid codes. This requirement is addressed by *subsets* specified using the *Reference Set* mechanism. In some cases, it is useful to express the range of possible values 'intensionally' by specifying rules rather than by listing every member of the set (e.g. to include all *concepts* that are *subtypes* of a specified *concept*).

The use of *postcoordinated expressions* adds further dimensions to the requirement for *constraints*. It may be necessary to specify whether all *postcoordinated refinements* of *concept* are permitted or whether some types of *refinement* are prohibited or required. It may also be necessary to specify whether a *postcoordinated expression* that is equivalent to a permitted value is itself permitted.

Requirements for representing *expression constraints* are closely related to the requirements for representing *predicates* in queries.

2.1.4.5 Query Predicates

Queries to be applied to *electronic health records* that including *SNOMED CT expressions* may need to represent predicates that test *postcoordinated expressions*. The requirements for representing *postcoordinated expression query predicates* are closely related to the requirements for representing *constraints* on *expressions*. While a *constraint* specifies whether a particular *expression* is permitted in a particular situation, an *expression* predicate specifies the range of candidate *expressions* that match the *query*.

2.2 Logical Abstract Models

This section provides a logical abstract view of *SNOMED CT components* and *derivatives*; and the use of these to represent instances of clinical information. Subsequent sections provide detailed technical *descriptions* of the *SNOMED CT components*, *derivatives* and related artifacts .

2.2.1 Logical Model of SNOMED CT Components

The abstract logical model of *SNOMED CT components* is illustrated by *Figure 8*. The model is centered around the representation of *concepts* and their associated *relationships* and *descriptions*.

Alignment between *release files* and the logical model:

- *SNOMED CT Release Format 2* is closely aligned with the logical model;
- *A mapping table* is provided with the *Release Format 1* file specification to map *RF1* file structures to the abstract model.

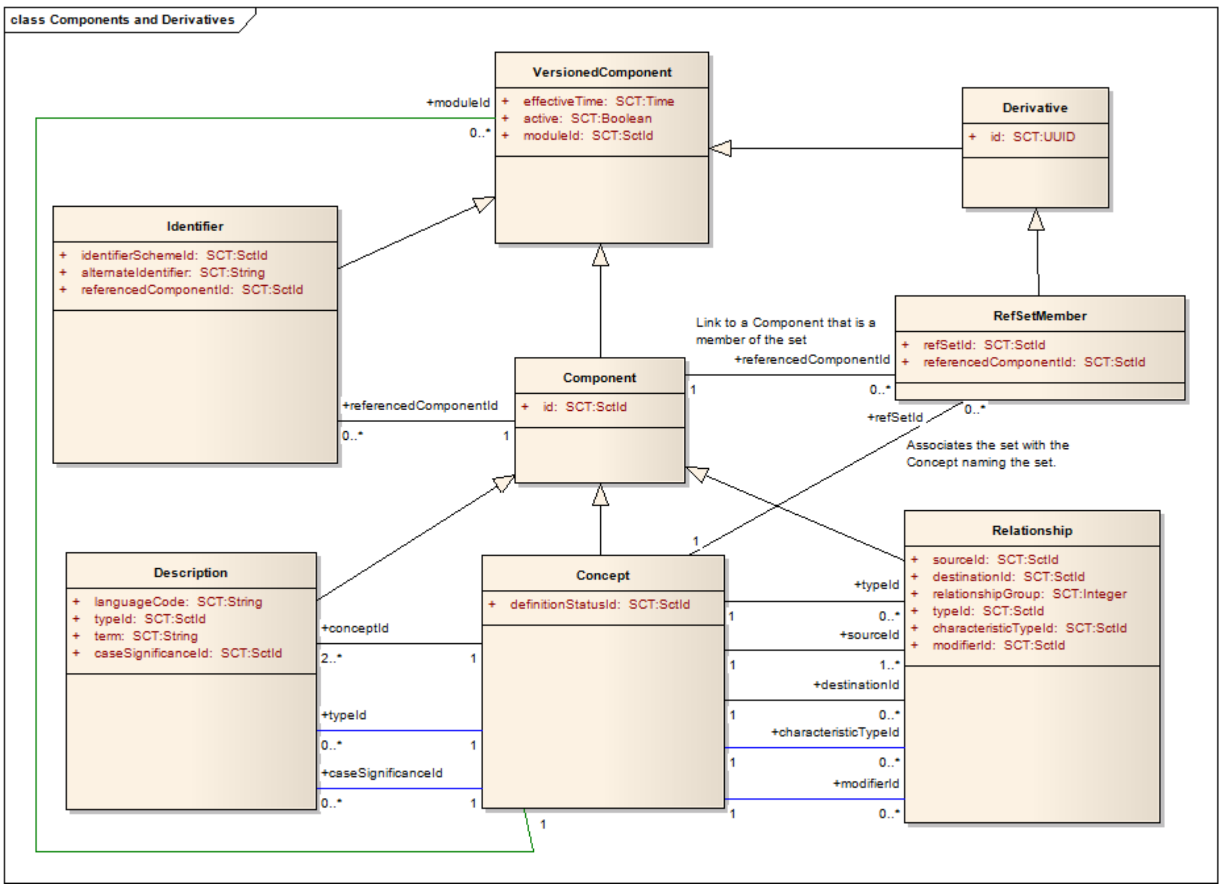


Figure 8: Abstract logical model of SNOMED CT components

2.2.1.1 Descriptions

The set of *terms* that describe a *concept*. These include *fully specified names*, *preferred terms* and *synonyms* in each supported *language*.

2.2.1.2 Relationships and concept definitions

Each *concept* is defined by a set of *relationships* to other *concept*. The resulting definition may be sufficient to distinguish the *concept* from its parents and siblings in the *subtype hierarchy* in which case the *concept* is considered to be *fully defined*. If the definition is not sufficient to distinguish the *concept* from its parents and siblings, the *concept* is said to be *primitive*. The *concept* contains a field that is set to indicate whether is definition status is *primitive* or *fully defined*.

Figure 9 illustrates the abstract logical model of a *concept*, including the defining *Relationships* between *concepts* (represented by the associations labelled *sourceId*, *destinationId* and *typeId*) and the definition status (represented by the *definitionStatusId*).

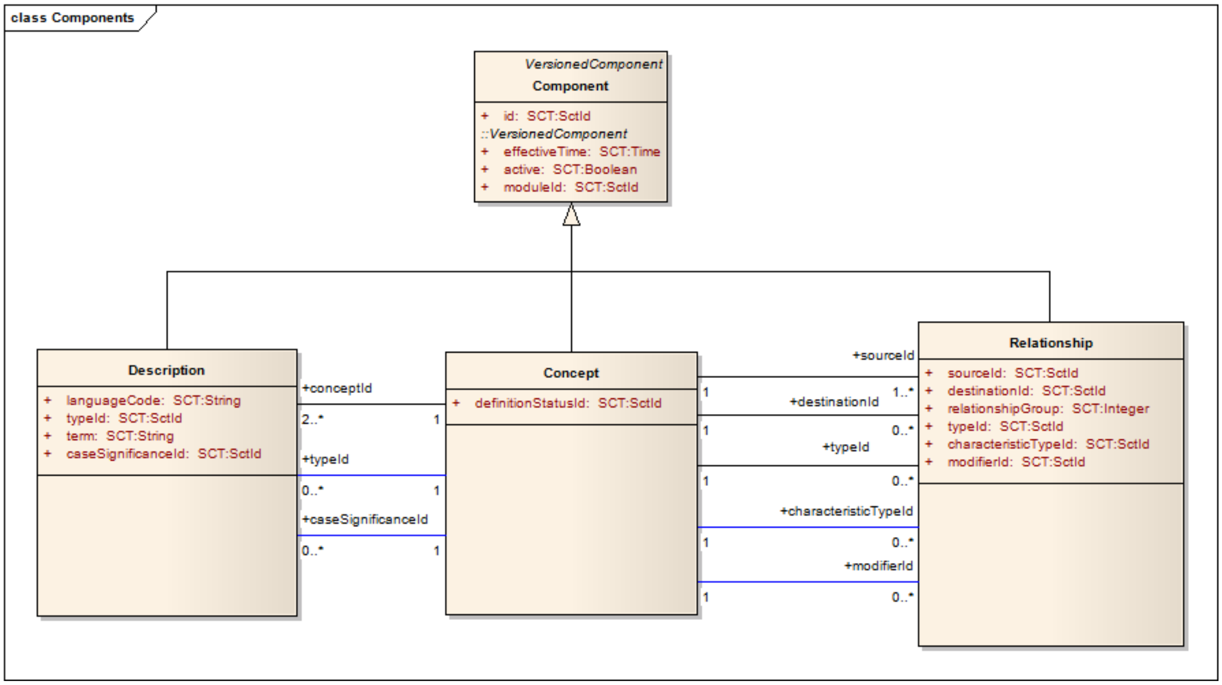


Figure 9: General Abstract Logical Model of a SNOMED CT concept definition

2.2.1.3 Alternative logical abstract model views of concept definitions

The definition of a *concept* can be logically transformed between different views without loss of meaning based on the definitions of related *concepts*.

For example:

Consider the following set of defining *relationships*:

pain in upper limb		is a		pain
pain in upper limb	has	finding site		upper limb structure
hand structure		is a		upper limb structure
Hand pain		is a		pain
Hand pain	has	finding site		hand structure

Based on the above five *relationships* it is possible to infer a new *relationship*:

| Hand pain | | is a | | pain in upper limb |

The definition of | Hand pain | can thus be viewed in three semantically identical forms:

1. As originally stated: :

- | Hand pain | | is a | | pain | and has | finding site | | hand structure |

or

2. With the additional inferred *relationship*:

   - | Hand pain | | is a | | pain | and | is a | | pain in upper limb | and has | finding site | | hand structure |

or

3. With the inferred *relationship* but without the redundant stated *relationship* | is a | | pain |: :

   - | Hand pain | | is a | | pain in upper limb | and has | finding site | | hand structure |

The *relationship* | is a | | pain | is redundant because this can be determined by traversing the | is a | *relationship* to | pain in upper limb | which in turn is defined as | is a | | pain |.

The result of manipulations like this is that several distinct views of the logical abstract model can be described based on the manner in which they are derived.

Different views of *concept* definitions vary in one or more of the following three dimensions:

- Flattened or nested;
- Stated or inferred;
- Direction and extent of logical *transformation*

These three dimensions are considered in the following subsections of this guide.

2.2.1.3.1 Flat and nested definition views

2.2.1.3.1.1 Flat definition views

In a flat view a *concept definition* consists only of defining *relationships* with target values that are themselves identified *concepts*.

To support this view *concepts* must be created (and defined) for any value that needs to be expressed in the definition of another *concept*.

Example: The | finding site | for the *concept* | pain in left hand | could only be defined by first creating a *concept* | structure of left hand | leading to a definition such as:

| pain in left hand | has | is a | | pain |.
| pain in left hand | has | finding site | | structure of left hand |.

The *concept* | structure of left hand | could be defined as follows:

| structure of left hand | | is a | | hand structure |
| structure of left hand | has | laterality | | left |.

2.2.1.3.1.2 Nested definition views

In a nested view of a *concept definition* the target value of a defining *relationship* may itself be a nested definition.

This avoids the need for creating intermediate *concepts* but results in more complex definitions.

Example:

The | finding site | for the *concept* | pain in left hand | could be defined without creating the *concept* | structure of left hand | by nesting an appropriate definition as follows:

| pain in left hand | | is a | | pain |
| pain in left hand | has | finding site | (| is a | | hand structure | and has | laterality | | left |).

2.2.1.3.1.3 SNOMED CT support for flat and nested definition views

Currently the *SNOMED CT* editing environment works with flat definition views and the standard relational distribution files do not support nested definition views.

Views of *concept definitions* that include nested definitions can be generated from existing *SNOMED CT* data. The proposed *SNOMED CT* XML distribution format does have the potential to support nested views.

Logically the flat form is as expressive as the nested form. The only difference is the need to create and define *concepts* to represent the nested elements in the definition.

Example:

To allow the *concept* | pain in left hand | to be *fully defined* without using a nested definition, | structure of left hand | must exist as a *concept* in *SNOMED CT*.

2.2.1.3.2 Stated and inferred definition views

2.2.1.3.2.1 Stated definition view

A stated *concept definition* is the set of *relationships* (and groups of *relationships*) that an author (*modeler*) has stated to be *defining characteristics* of a *concept*. The *stated view* is maintained in the *SNOMED CT* editing environment and is reviewed and modified during the process of editing a revised edition of *SNOMED CT*.

The *stated view* is distributed in a format similar to the *relationship file*.

2.2.1.3.2.2 Inferred definition views

Inferred *concept definitions* are derived from a stated *concept definition* taking account of the definitions of the *concepts* referred to in the stated definition.

Inferences are derived by applying a consistent set of logical rules to the definition taking account of the definitions of related *concepts*.

Several semantically identical views may be inferred and these are discussed in the following section.

The standard *SNOMED CT* distribution includes the *relationships table* which represents one of the inferred views of the definitions of all *active concepts*.

2.2.1.3.2.3 Alternative inferred definition views

Several semantically identical views may be inferred by applying different logical *transformations* to the *stated view*. Logical *transformations* may vary in the extent to which they normalize the definition and the level of redundancy in the resulting definition.

Different inferred views have properties that optimize different types of function.

The extreme points in the spectrum of possible *concept* definition views are:

- Comprehensive:
  - The set of all defining *relationships* that can be inferred to be true for a *concept* based on the stated definition of this *concept* and the stated definitions of all other directly or indirectly related *concepts*.
- Minimal:
  - The smallest set of defining *relationships* that expresses the definition of the *concept*.

Each inferred view is a combination of a specific *supertype view* (| is a | *relationships*) and an *attribute view* (other defining *relationships*).

2.2.1.3.2.3.1 Supertype aspects of inferred definition views

An inferred definition view includes one of several alternative views of the supertype | is a | *Relationships*. The considerations in this section exclude the *defining characteristics* of a *concept*.

2.2.1.3.2.3.1.1 Comprehensive view of supertype ancestors ("transitive closure")

An inferred *concept definition* view may explicitly contain *relationships* to all supertypes *ancestors* of the defined *concept*.

This comprehensive view of supertypes is known in *description logic* as a "*transitive closure*". It involves traversing (transiting) the target of each |is a| *relationship* to look for and follow further |is a| *relationships* until all paths through the *hierarchy* reach the *root concept* (closure).

This is a highly redundant *expression* of the logical abstract model of a *concept* definition. Applied to the full content of *SNOMED CT* it results in tens of millions of *relationships*.

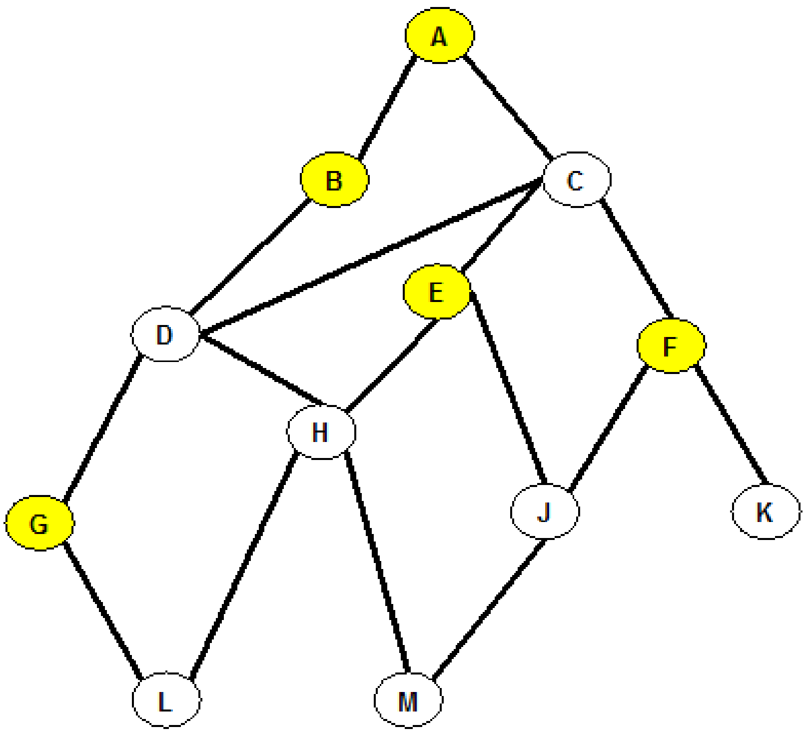


Figure 10: Example hierarchy with list of supertypes in the transitive closure

Table 6: Transitive Closure of Supertypes in the Example Hierarchy

Concept	***Transitive closure* of supertypes**
A	-
B	A
C	A
D	A, B, C
E	A, C
F	A, C
G	A, B, C, D

Concept	***Transitive closure* of supertypes**
H	A, B, C, D, E
J	A, C, E, F
K	A, C, F
L	A, B, C, D, E, G, H
M	A, B, C, D, E, F, H, J

The advantage of this type of view is that there is no need to walk the *hierarchy* tree to answer the question "is *concept* M subsumed by *concept* B". Instead this can be answered simply by checking the *transitive closure* of " *concept* M" for the presence of " *concept* B". Therefore, this view enables high-performance subsumption testing.

Note: Experience suggests that a pre-computed *transitive closure* table out-performs other options and is robust, flexible and easy to implement. Therefore, unless storage capacity is significant concern, this approach is recommended.

2.2.1.3.2.3.1.2 Proximal supertype view (standard distribution view)

An inferred view of a *concept definition* may contain *relationships* to the set of proximate *supertype parents* of that *concept*. *Relationships* with other *supertype ancestors* that can be reached by traversing multiple **|** is a **|** *relationships* are omitted.

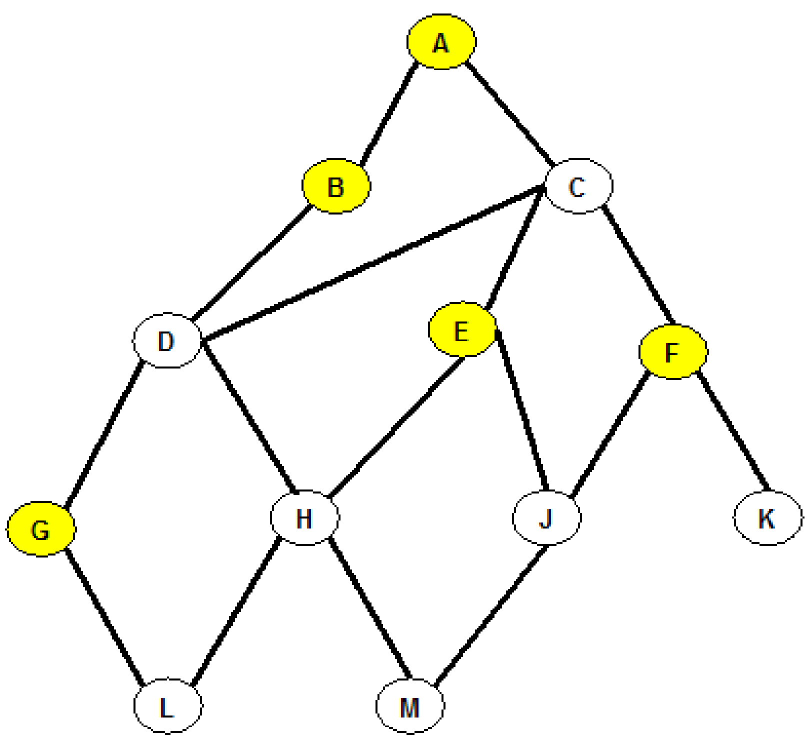


Figure 11: Example hierarchy with list of proximal supertypes

Table 7: Proximal Supertypes in the Example Hierarchy

Concept	**List of proximal supertypes**
A	-
B	A
C	A
D	B, C
E	C
F	C
G	D
H	D, E
J	E, F
K	F
L	G, H
M	H, J

2.2.1.3.2.3.1.3 Comprehensive primitive supertype view

An inferred view of a *concept definition* may contain *relationships* to all *supertype ancestors* that are "*primitive*" *concepts* (yellow shaded in examples).

The rationale for this is that all the distinguishing features of the "*fully defined*" *concepts* (white unshaded in examples) are represented by other defining *relationships* which will show up in the attribute part of the view.

This view can be used when testing whether a candidate *concept* is subsumed by a predicate *expression*. If the proximal *primitive* supertype view of the predicate *expression* includes any *concept* that is not in the comprehensive *primitive* view of the candidate *concept definition*, then the *concept* is not subsumed by the *expression*.

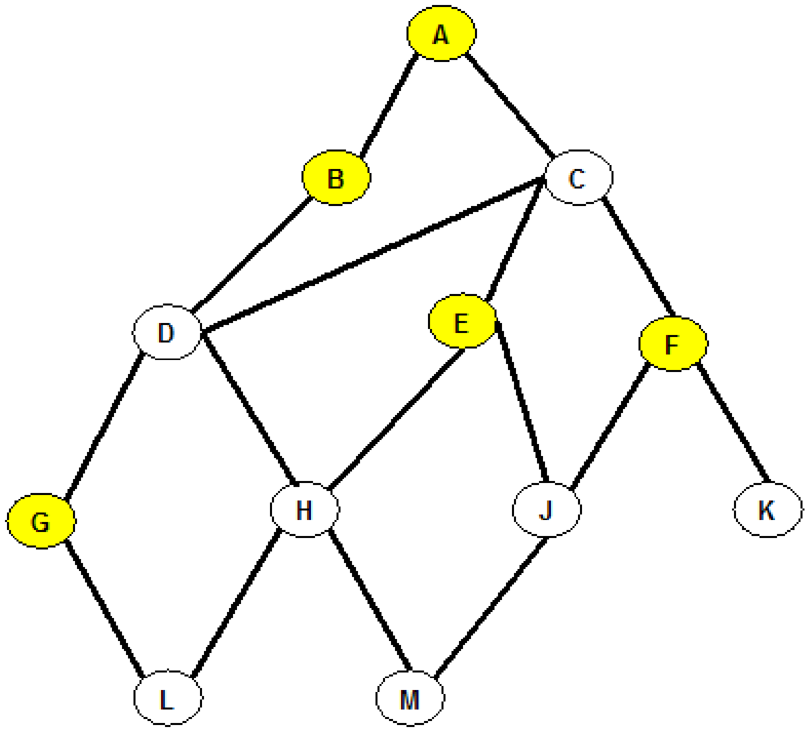


Figure 12: Example hierarchy with comprehensive list of primitive supertypes

Table 8: Primitive Supertypes in the Example Hierarchy

Concept	**Comprehensive list of *primitive* supertypes**
A	A
B	A, B
C	A
D	A, B
E	A, E
F	A, F
G	A, B, G
H	A, B, E
J	A, E, F
K	A, F
L	A, B, E, G

Concept	**Comprehensive list of *primitive* supertypes**
M	A, B, E, F

Note:

1. In this view the definitions of *primitive concepts* should implicitly or explicitly include a reference to the defined *concept* itself. This is because a *primitive concept* expresses some meaning that is not fully distinguished from its supertypes by other defining *relationships*. The reference to self need not be explicitly stored and provided that it is included implicitly at run time.
2. All *active concepts* include the *root concept* in their *transitive closure*. The reference to root need not be explicitly stored provided that it is included implicitly at run time.

2.2.1.3.2.3.1.4 Proximal primitive supertypes (short normal view)

An inferred *concept definition* may contain *relationships* to the set of proximate *primitive supertype parents* of that *concept*. *Relationships* with *fully defined supertype ancestors* are omitted as are *relationships* with *primitive ancestors* that are also supertypes of one of proximate *primitive* supertypes.

This view can be used to test if a candidate *expression* is subsumed by a predicate *concept*. If the proximal *primitive* supertype view of the *concept definition* of the predicate includes any *concept* that is not in the comprehensive *primitive* view of the candidate *expression*, then the *expression* is not subsumed by the *concept*.

The **|** is a **|** *relationships* in the *SNOMED CT* '*canonical table*' represent this view.

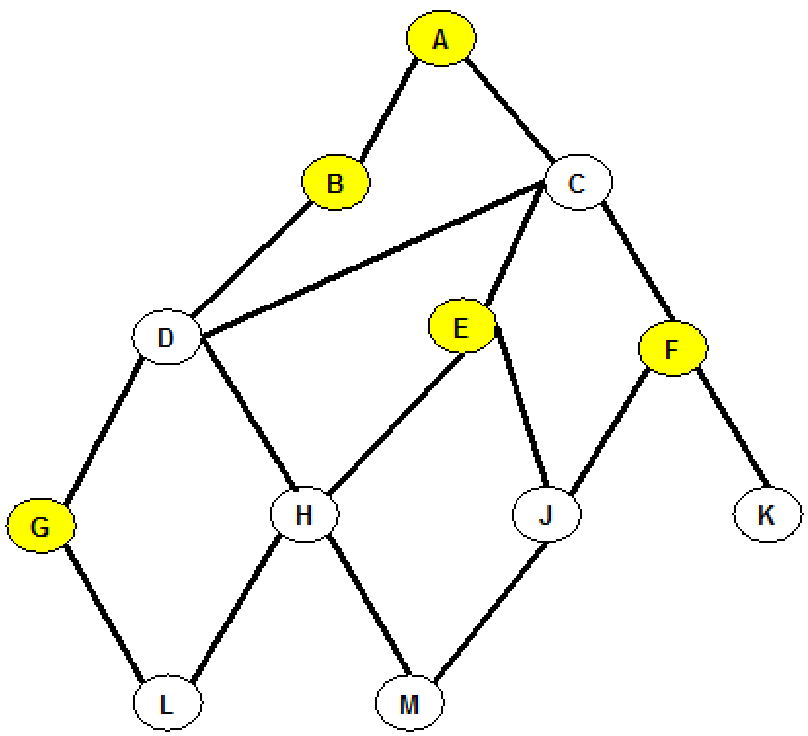


Figure 13: Example hierarchy with list of proximal primitive supertypes

Table 9: Proximal Primitive Supertypes in the Example Hierarchy

Concept	List of proximal *primitives*
A	A
B	B
C	A
D	B
E	E
F	F
G	G
H	B, E
J	E, F
K	F
L	E, G
M	B, E, F

Note: The proximal *primitive* of a *primitive concept* is the *concept* itself.

2.2.1.3.2.3.2 Attribute aspects of concept definition views

An inferred definition view includes one of several alternative views of the *defining characteristics* of a *concept*. The considerations in this section exclude the supertype | is a | *relationships*.

In addition to the different views described in this section, alternative logical forms may be applied to the values of the *relationships*.

2.2.1.3.2.3.2.1 Comprehensive view of defining Relationships

An inferred *concept definition* may include all the defining *relationships* (and *relationships* groups) that are known to be true. This includes those stated and other inferred by inheritance from stated *supertype ancestors*.

The full form includes all possible *supertype ancestor* values of the stated attributes. This means that in many cases this will include a very large set of *relationships*.

Taken to its logical extreme this also includes *relationships* duplication of *relationships* with *relationship types* that are supertypes of those types stated (e.g. all | procedure site - indirect | *relationships* would be duplicated for the supertype attribute | procedure site |).

While this version of the definition model is an Abstract Logical view it is unlikely that explicit representation of this view will deliver benefits sufficient to merit this level of redundancy.

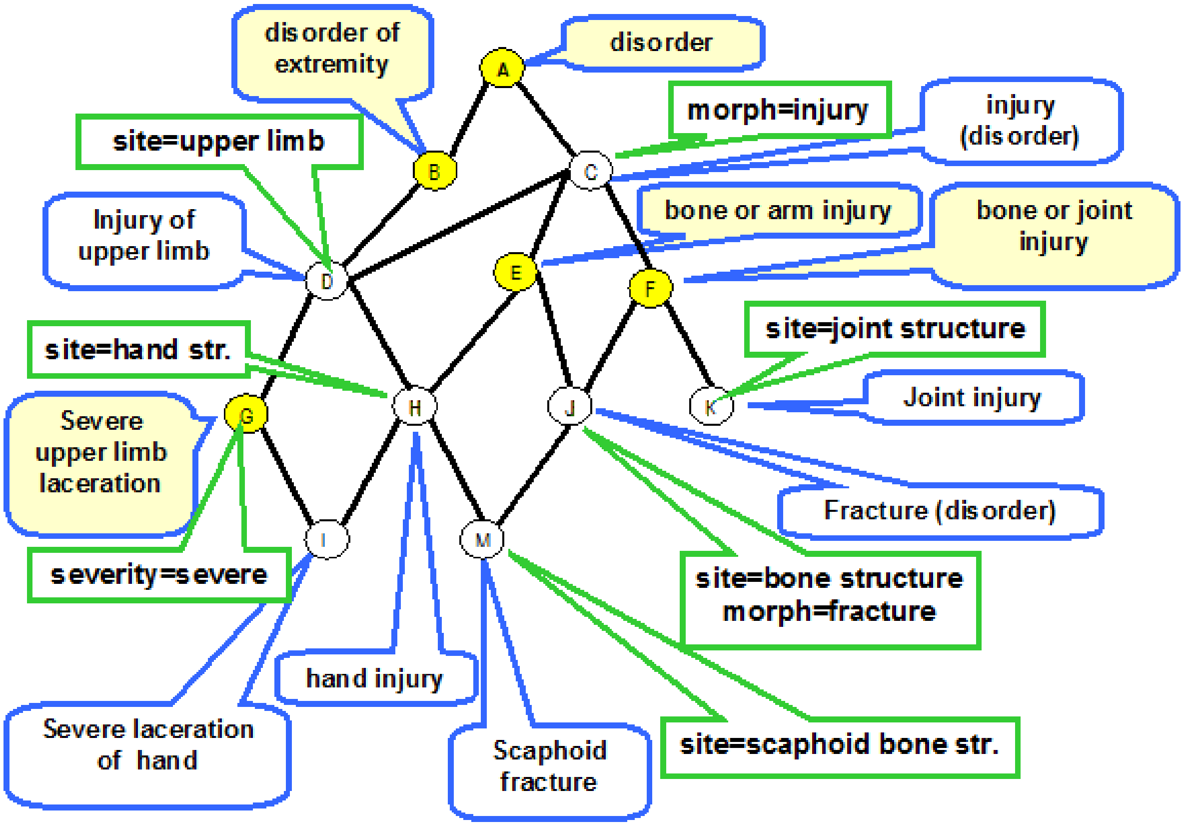


Figure 14: Illustration of sample concepts with differentiating defining characteristics

Table 10: Comprehensive attribute view of sample concepts

See *Figure 14*

C. | Injury disorder | | morphology |=| injury |	**D. | Injury of upper limb |** | site |=| upper limb structure | | morphology |=| injury |
E. | Bone or arm injury | *(primitive)* | morphology |=| injury |	**F. | Bone or joint injury |** *(primitive)* | morphology |=| injury |
G. | Severe upper limb laceration | *(primitive)* | site |-upper limb structure | morphology |=| injury | | severity |=| severe |	**H. | Hand injury |** | site |=| upper limb structure | | site |=| hand structure | | morphology |=| injury |
J. | Fracture (disorder) | | site |=bone structure | morphology |=| injury | | morphology |=| fracture |	**K. | Joint injury |** | site |=| joint structure | | morphology |=| injury |

L. | Severe laceration of hand | | site |=| upper limb structure | | site |=| hand structure | | morphology |=| injury | | severity |=| severe | <u>Note</u> Although the morphology | laceration | is not specified in the example | severe upper limb laceration | refined to the site hand fully defines this *concept*. In a complete view (including supertypes and attributes) this difference is clear.	**M. | Scaphoid fracture |** | site |=| upper limb structure | | site |=| hand structure | | site |=| bone structure | | site |=| scaphoid bone structure | | morphology |=| injury | | morphology |=| fracture |

2.2.1.3.2.3.2.2 Non-redundant defining Relationships ("distribution view")

An inferred *concept definition* may include the set of non-redundant defining *relationships* (and *relationships* groups) that are known to be true. This includes those stated and others inferred by inheritance from stated *supertype ancestors*. However, any *relationships* (or *relationships* groups) that are supertypes of other *relationships* (or *relationship groups*) are redundant and are not included in this view.

A *relationship* that is part of a *relationship group* is only regarded as redundant if the *relationship group* as a whole subsumes another *relationship group*.

This is the view expressed in the standard *SNOMED CT* distribution and this same view also forms part of the long *normal form*.

Table 11: Non-redundant attribute views of sample concepts

See *Figure 14*

C. | Injury disorder | | morphology |=| injury |	**D. | Injury of upper limb |** | site |=| upper limb structure | | morphology |=| injury |
E. | Bone or arm injury | *(primitive)* | morphology |=| injury |	**F. | Bone or joint injury |** *(primitive)* | morphology |=| injury |
G. | Severe upper limb laceration | *(primitive)* | site |=| upper limb structure | | morphology |=| injury | | severity |=| severe |	**H. | Hand injury |** | site |=| hand structure | | morphology |=| injury |
J. | Fracture (disorder) | | site |=| bone structure | | morphology |=| fracture |	**K. | Joint injury |** | site |=| joint structure | | morphology |=| injury |

L. | Severe laceration of hand | | site |=| hand structure | | morphology |=| injury | | severity |=| severe | Note Although the morphology | laceration | is not specified in the example | severe upper limb laceration | refined to the site hand fully defines this *concept*. In a complete view (including supertypes and attributes) this difference is clear.	**M. | Scaphoid fracture |** | site |=| scaphoid bone structure | | morphology |=| fracture |

2.2.1.3.2.3.2.3 Primitive differential attribute view of concept definitions

The *primitive* differential view includes only non-redundant defining *relationships* (and *relationship groups*) that are not present in the sum of the definitions of the set of *primitive* supertype *concepts*. This view provides a minimal attribute view which is semantically complete when combined with one of the *primitive* supertype views.

A *relationship* that is part of a *relationship group* is only regarded as redundant if the *relationship group* as a whole subsumes another *relationship group*.

Table 12: Primitive differential attribute views of sample concepts

See *Figure 14*

C. | Injury disorder | | morphology |=| injury |	**D. | Injury of upper limb |** | site |=| upper limb structure | | morphology |=| injury |
E. | Bone or arm injury | *(primitive)* (| morphology |=| injury |) note 2	**F. | Bone or joint injury |** *(primitive)* (| morphology |=| injury |) note 2
G. | Severe upper limb laceration | *(primitive)* (| site |-| upper limb structure | | morphology |=| injury | | severity |=severe) | note | 2	**H. | Hand injury |** | site |=| hand structure | | morphology |=| injury |
J. | Fracture (disorder) | | site |=| bone structure | | morphology |=| fracture |	**K. | Joint injury |** | site |=| joint structure | | morphology |=| injury |
L. | Severe laceration of hand | | site |=| hand structure |	**M. | Scaphoid fracture |** | site |=| scaphoid bone structure | | morphology |=| fracture |

Note:

1. This is the attribute view expressed in the *SNOMED CT canonical form* table.
2. If the *primitive* supertype view of *primitive concepts* includes the *concept* itself (i.e. as its own proximal *primitive*) then the differential attribute view is empty for all *primitive concepts*. The entries shown above for *primitive concept* apply only where the *concept* itself is excluded from the proximal *primitive* supertype view.

2.2.1.3.2.3.2.4 Supertype differential attribute view of concept definitions

The supertype differential view includes only non-redundant defining *relationships* (and *relationship groups*) that are not present in the sum of the definitions of the supertypes of the *concept*. This view provides a minimal attribute view which is semantically complete when combined with the proximal or complete supertype view.

A *relationship* that is part of a *relationship group* is only regarded as redundant if the *relationship group* as a whole subsumes another *relationship group*.

Table 13: Supertype differential attribute views of sample concepts

See *Figure 14*

C. | Injury disorder | | morphology |=| injury |	**D. | Injury of upper limb |** | site |=| upper limb structure | | morphology |=| injury |
E. | Bone or arm injury | *(primitive)*	**F. | Bone or joint injury |** *(primitive)*
G. | Severe upper limb laceration | *(primitive)* | severity |=| severe |	**H. | Hand injury |** | site |=| hand structure |
J. | Fracture (disorder) | | site |=| bone structure | | morphology |=| fracture |	**K. Joint injury** | site |=| joint structure |
L. | Severe laceration of hand | *None* Note All distinguishing characteristics are inherited from one or both of the supertypes.	**M. | Scaphoid fracture |** | site |=| scaphoid bone structure |

2.2.1.3.2.3.3 The Short Canonical Form

The short *canonical form* is an alternative view of the *Relationships* that is provided as an RF1 *release file*. It consists of the union of the following two views:

- *Proximal primitive supertypes (short normal view)*
- *Primitive differential attribute view of concept definitions*.

2.2.1.3.3 Nature of the definition

A *concept definition* has one of the following two forms:

1. *fully defined concepts* :
    - The definition is complete. It contains *relationships* that represent the full set of *necessary* and *sufficient* conditions.
2. *primitive concepts* :
    - The definition is incomplete. It contains *relationships* that represent a set of *necessary* conditions but this set of conditions is not *sufficient* to fully define the *concept*.

Note: A *necessary* condition is a characteristic that is always true of a *concept*.

Example: | morphology | = | fracture | is a necessary condition of | fracture of femur |.

Note: If all members of a *sufficient* set of conditions are true they imply that the *concept* is also true.

Example: | morphology | = | fracture | and | finding site | = | bone structure of femur | form a *sufficient* set of conditions that define the *concept* | fracture of femur |.

Note: All members of the set of sufficient conditions are also necessary conditions. However, some *necessary* conditions may not form part of the *sufficient* set of conditions.

Example:

Consider the *concept* | gastric ulcer |

- The | finding site | = | gastric mucosa | is a *necessary* condition for | gastric ulcer |:
    - This is true because all gastric ulcers necessarily involve the | gastric mucosa |
- The definition | morphology | = | ulcer | and | finding site | = | stomach structure | is a *sufficient* definition for | gastric ulcer |:
    - This is true because any ulcer in a stomach structure is a | gastric ulcer |.
- Therefore, an assertion that a person has an | ulcer | with | finding site | | stomach | is *sufficient* to imply that they have a | gastric ulcer |:
    - Since a gastric ulcer *necessarily* involves the | gastric mucosa | it should be possible to deduce that a person with an "ulcer" with finding site | stomach | has a disorder of with a site | gastric mucosa |.

2.2.1.3.3.1 Sufficient definition

A *sufficient* definition consists of a set of defining *relationships* (and *relationship groups*) which taken together imply a particular meaning.

The value of a *sufficient* definition is that it allows post coordinated *expression* that is sufficient to define a *concept* to be recognized as equivalent to (or a *subtype* of) a defined *concept*.

For example:

Gastric ulcer is defined as follows and this is a *sufficient* definition because any | ulcer | in a | stomach structure | is by definition a | gastric ulcer |.

116680003 | is a | =64572001 | disease | {116676008 | associated morphology | =56208002 | ulcer | ,363698007 | finding site | =69695003 | stomach structure |}

Based on this definition:

Any *postcoordinated expression* that specified a disease involving an | ulcer | with | finding site | | stomach | would be equivalent to or a *subtype* of | gastric ulcer |.

However, a *query* for all disorders involving | gastric mucosa | would incorrectly exclude the *concept* | gastric ulcer | as the site is not specified as some stomach structure rather then specifically identifying the gastric mucosa.

2.2.1.3.3.2 Necessary definition

A *necessary* definition consists of a set of defining *relationships* (and *relationship groups*) which express all the attributes that are necessarily true about a *concept* for a given version of the *SNOMED CT Concept Model*.

A *necessary* definition may contain *relationships* or *refinements* that are not essential for a *sufficient* definition.

The value of a *necessary* definition is that it allows more refined subsumption queries to be appropriately evaluated.

For example:

Gastric ulcer could be defined as follows:

116680003 | is a | =64572001 | disease | {116676008 | associated morphology | =56208002 | ulcer | ,363698007 | finding site |78653002 | gastric mucous membrane structure |}

This more tightly defined definition contains a *necessary* definition (| finding site | = | gastric mucous membrane structure |). This is necessarily true if the sufficient definition (| finding site | = | stomach structure |) is true, because any ulcer in a stomach structure is by definition a gastric ulcer.

2.2.1.3.3.3 Limitations of the current SNOMED CT model

The *current SNOMED CT* model and distribution format do not distinguish between *relationships* that are *necessary conditions* and those that are part of a set of *necessary and sufficient* conditions. For any *fully defined concepts* the set of defining *relationships* are regarded as *necessary and sufficient*.

As a result of this limitation some currently released *fully defined concept* definitions may include conditions that are *necessarily* true but are not required as part of the set of *sufficient* conditions.

Example: Consider the two definitions shown below:

116680003 | is a |=64572001 | disease | , 246075003 | Causative agent |=113858008 | mycobacterium tuberculosis complex | { 116676008 | associated morphology |=6266001 | granulomatous inflammation | , 363698007 | finding site |=39352004 | joint structure |}

Figure 15: | tuberculous arthritis |

116680003 | is a |=64572001 | disease | , 246075003 | causative agent |=41146007 | bacteria | { 116676008 | associated morphology |=23583003 | inflammation |, 363698007 | finding site |=39352004 | joint structure |}

Figure 16: | bacterial arthritis |

The definition of | tuberculous arthritis | differs from that of | bacterial arthritis | in two respects. In practice the first of these (| causative agent |= | mycobacterium tuberculosis complex |) is sufficient to define the *concept*. However, the nature of the inflammation that results is, necessarily, granulomatous.

Thus an *expression* that specifies | bacterial arthritis | with | causative agent |=| mycobacterium tuberculosis complex | is clinically equivalent to the *concept* | tuberculous arthritis | even though it does not explicitly refine the nature of the inflammation.

In contrast the current *SNOMED CT* model computes | bacterial arthritis | with | causative agent |=| mycobacterium tuberculosis complex | as supertype of | tuberculous arthritis |. This occurs because the *expression* | bacterial arthritis | with | causative agent |=| mycobacterium tuberculosis complex | does not specify of the nature of the inflammation.

Future enhancements: Options for distinguishing the sufficient set of defining *relationships* from those that are merely necessarily true are being investigated. A complete solution to this issue needs to support the recognition of several separate sufficient sets. However, initially a solution recognizing a single sufficient set may be introduced.

2.2.1.3.3.4 Impact on retrieval

A *necessary* definition is inevitably more complete than a *sufficient*definition. From the perspective of retrieval the completeness of a definition is a mixed blessing.

- It is an advantage for candidate *expressions* as they will be subsumed by a wider set of appropriate predicates.
- It is a disadvantage for a predicate *expression*, the necessary conditions may result in incomplete retrieval. A candidate *expression* that satisfies all the *sufficient* conditions should be included. However, it will be excluded unless it satisfies all the necessary conditions in the predicate.

This occurs where the definition of a *concept* states conditions that are *necessarily* true but which go beyond those that are *sufficient* to distinguish a *concept* from its supertypes.

Example:

The *normal form* definition of | pulmonary tuberculosis | is as follows:

116680003 | is a | = 64572001 | disease |

,246075003 | causative agent | = 113858008 | mycobacterium tuberculosis complex |

{116676008 | associated morphology | = 6266001 | granulomatous inflammation |

,363698007 | finding site | = 39607008 | lung structure | }

Used as a *query* predicate, this will exclude valid candidate *expressions* such as ...

233604007 | pneumonia | : 246075003 | causative agent | = 113861009 | mycobacterium tuberculosis |

- This *expression* is not subsumed by the full definition of | pulmonary tuberculosis | because it does not mention "granulomatous inflammation". This type of inflammation is characteristic of "mycobacterium tuberculosis" infection and so is necessarily present. Since currently *SNOMED CT* definitions do not distinguish the sufficient and necessary conditions this cannot be inferred.

A more inclusive *query* predicate that specifies a sufficient set of conditions for | pulmonary tuberculosis | can be constructed by removing the morphology condition.

116680003 | is a | = 64572001 | disease |

,246075003 | causative agent | = 113858008 | mycobacterium tuberculosis complex |

,363698007 | finding site | = 39607008 | lung structure |

- This correctly subsumes both the *precoordinated concept* | pulmonary tuberculosis | and the *postcoordinated* candidate *expression* above.

Note: To ensure complete retrieval

- When selecting a *concept* as part of a *query* predicate, view its *normal form* definition and decide whether some of the conditions should be omitted;
- Specify the minimum set of conditions sufficient for the intended purpose.

Future enhancements: In future, when the *SNOMED CT* model is revised to distinguish *sufficient* sets of defining *Relationships*, the sufficient definition can be used as the predicate for a retrieval. A candidate *expression* matches a predicate if it *necessarily* fulfills all the*sufficient* conditions specified in the *query*.

2.2.1.4 Non-definitional Relationships

The model used to specify *SNOMED CT* defining *relationships* is also used to express non-defining *relationships* between *concepts*. These are distinguished by the CharacteristicTypeId attribute.

2.2.1.4.1 Qualifiers

Qualifiers specify a named *relationship* that may be used to refine the meaning of a *concept*. *Qualifiers* specify the allowable value *concept* (s) that can be applied to refine the *concept*.

Qualifiers are not part of the definition of a *concept*. However, *qualifiers* may be applied to a *concept* as part of an *expression* to refine the meaning of the *concept*.

Note: *Qualifiers* only cover a some of the permitted refinements that can be made to *concepts*. It is likely that in future *qualifiers* will superseded by use of a more complete machine-processable representation of the range of refinements permitted by the *Concept Model*.

2.2.1.4.2 Additional Relationships

Additional *relationships* represent characteristics of a *concept* that are not part of the definition of the *concept*.

Additional *relationships* are typically relevant to a particular national or organizational context of use and they may also change over time.

Example: Within the *UK NHS* the legal prescribing status of drugs are represented using additional *Relationships*. The fact that a particular drug can only be obtained with a prescription is indicated by *Relationships* such as the following:

- | amoxicillin 250 mg capsules | has | NHS prescribing status messages | = | prescription only medication |

2.2.1.4.3 Historical relationships

Historical relationships are used to relate *inactive concepts* to *active concepts*.

Historical relationships provide a bridge between *concepts* that are now *inactive* (and thus not formally defined) and *concepts* that are *active*. This allows pre-existing records which use the now *inactive concept* to be appropriately interpreted.

Note: *Historical relationships* are replaced by *Historical Association Reference Sets* in *Release Format 2*.

2.2.1.5 Other non-definitional properties of concepts

2.2.1.5.1 Legacy codes

Legacy *Identifiers* present in the *SNOMED CT* logical model include the *CTV3ID* (the *Read Code* from *NHS Clinical Terms Version 3*) and the SnomedId (the *SNOMED* code used in *SNOMED* 3).

Note: The *CTV3ID* and *SNOMEDID* fields are no longer supported in *Release Format 2*. Instead a |*Simple map (reference set)*| is used to document the link between legacy codes and *SNOMED CT*.

Note: The *CTV3ID* field should no longer be relied upon for mapping to and from the *Read Codes*. Additional mapping work in the UK identified some anomalies and resulted development of more flexibility table for *Read Code* Mapping

2.2.1.5.2 Cross Maps

Cross Maps to other terminologies and classifications are indirectly a part of the logical *SNOMED CT* model for *concepts*. However, this aspect of the model is outside the scope of this guide.

2.2.2 Logical Model of SNOMED CT expressions

Figure 17 shows the general abstract model for a *SNOMED CT expression*. This diagram also shows the references between *expressions* and components.

An *expression* is a collection of references to one or more *concepts*. The *expression* consists one or more *focus concepts* and an optional *refinement*.

The *focus concept* and the names of the refining attributes are represented by references to *SNOMED CT concepts*. The value of a refining attribute is itself an *expression* and is structured in the same way. Thus nested *expression* can be used to refine the value of a refining attribute.

An *expression* represents an instance of the meaning defined by the defining *relationships* of the *focus concepts* as modified by the *refinements*.

The meaning of each *refinement* is expressed by an *attribute name* which names a property and an *attribute-value* which expresses the value of that property.

- The *attribute name* must be a *concept* that is a *subtype* of |attribute|.
- The *refinement* value may be a *concept* or *expression* that is a appropriate to the named attribute. The values that are appropriate to an attribute are specified by the *Concept Model*. In most cases, any *subtype* of a *concept* that is permitted as a value of an attribute is also permitted.
- *Refinements* may be grouped to represent interdependencies between them in the same way as *relationship groups*.

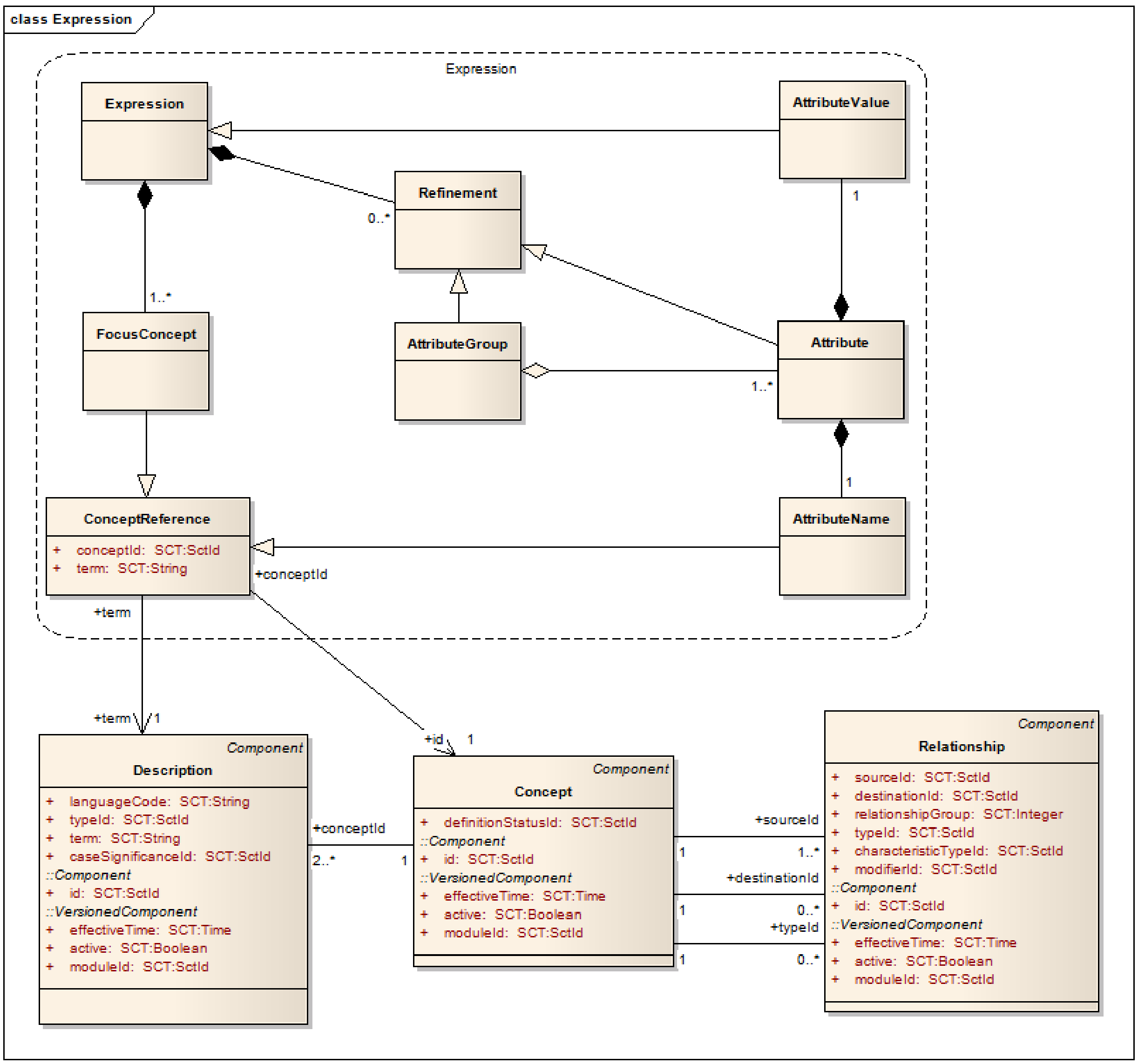


Figure 17: General Abstract Logical Model of a SNOMED CT expression

2.2.2.1 Refinement

An *expression* represents an instance of the meaning defined of the *focus concepts* are as modified by the *refinements*.

Various types of *refinement* are possible. Of these some are fully supported by the *SNOMED CT Concept Model* and released data while other possible methods of *refinement* step outside those boundaries.

2.2.2.1.1 Refinement of defining Relationships

2.2.2.1.1.1 Refinement individual attribute values

A defining *relationship* of the base *concept* can be refined by applying a value that is a *subtype* of the defining value.

This approach to *refinement* is fully supported by the *SNOMED CT Concept Model*.

Defining *relationships* that are marked with the *refinability* property value "not refinable" should not be refined.

2.2.2.1.1.2 Refinement attribute names

A defining *relationship* of the base *concept* can also be refined by applying a name that is a *subtype* of the defining *attribute name*. For example, if the defining *relationship* specifies a | procedure site | this may be refined to | procedure site - direct | or | procedure site - indirect |.

2.2.2.1.1.3 Refinement of defining Relationship groups

If a *refinement* is applied to one of the defining *relationships* within a *relationship group*, it is the group a whole that is refined.

It is also permissible for a stated (close-to-user) *expression* to refine a *relationship* without grouping the refined *relationship* or without fully enumerating the group of which it is part. In this case, resolution to an inferred structure should apply the ungrouped *relationship* value (or partially enumerated group) as a *refinement* of any group to which that *refinement* can be appropriately applied.

2.2.2.1.1.4 Nested refinement of defining Relationships

The value of a defining *relationship* may itself be refined. In this case the value of the *relationship* becomes a *postcoordinated expression* rather than a *precoordinated concept*.

This occurs most frequently in the following situations:

Laterality refinement

The laterality qualification applies to the value of the | procedure site | or | finding site | *relationship* and is logically nested under site.

(Note lateralization is discussed separately)

Refinement of *situation with explicit contexts*

The | associated finding | or | associated procedure | is a | clinical finding | or | procedure |, which may itself be refined (e.g. with severity).

2.2.2.1.2 Applying values to qualifiers

2.2.2.1.2.1 Applying values to individual qualifiers

A qualifying *relationship* of the base *concept* can be used to apply a *refinement*.
The nature of the allowable *refinement* using *qualifiers* is determined by the value of the "*refinability* property" of the qualifying *relationship*.

<u>Not refinable</u>

The *qualifier* can only be used to refine the base *concept* by applying the qualifying value specified in the distributed table.

<u>Refinable</u>

The *qualifier* can be used to refine the base *concept* by applying the qualifying value specified in the distributed table or any *subtype* of that value.

Mandatory to refine

The *qualifier* can be used to refine the base *concept* by applying a *subtype* of the qualifying value specified in the distributed table. It cannot be applied with the specified value itself as this is a non-specific grouping value for possible *refinements*.

This approach to *refinement* is fully supported by the *SNOMED CT Concept Model*.

2.2.2.1.2.2 Grouping qualifier refinements

In theory the value of a *qualifier* may apply only to the content of one *relationship group*.

Currently *qualifiers* are not grouped in *SNOMED CT releases* and therefore grouping of *qualifier refinements* is not supported in the *current Concept Model*. However, this is under review and the model may be extended to include grouped *qualifiers* in future. This review is required because problems arise with subsumption testing where *precoordinated* definitions include grouped *attribute-value pairs* and the *expression* uses an ungrouped, *qualifier* derived, attribute.

2.2.2.1.2.3 Nested refinement of qualifiers

The value of a *qualifier* may itself be refined and represented as an *expression* rather than a *precoordinated concept*.

This occurs most frequently with *expressions* which qualify high level "*situation with explicit context*"*concepts* (e.g. "finding with *explicit context*"). In this case the | associated finding | is applied as a *qualifier* which may itself be refined (e.g. with severity).

2.2.2.1.3 Applying laterality to a concept

A laterality value (left, right or bilateral) can be applied as a *qualifier* to lateralizable body structure *concepts*.

It is also permissible for a stated (close-to-user) *expression* to lateralize a base *concept* that has a definition including reference to a lateralizable body structure. In this case, resolution to an inferred structure should apply the laterality to all values in the base *concept definition* that are lateralizable body structures.

This approach is fully supported by the *SNOMED CT Concept Model*, provided that appropriate *transforms* are applied.

Note

If lateralization is specific to particular aspects of the *concept* then the laterality should be applied to the appropriate *relationship* as part of a nested *expression*.

2.2.2.1.4 Sanctioned and unsanctioned refinement

2.2.2.1.4.1 Introduction to refinement sanctioning

SNOMED CT relationships provide information that may be used to determine the types of *refinements* can be processed to determine *equivalence* and subsumption. However, even where a *concept* has no specific *relationship* it is possible to apply a refinement using an *attribute* that the *Concept Model* permits for *concepts* in that domain. Other *attributes* are not recommended for refinement as they will result in *expression* that cannot be normalized or reliably compared. Specific issues with unsanctioned refinements are considered in:

- *Unsanctioned use of "Concept Model attributes"*;
- *Use of "unapproved attributes"*;
- *Advantages and disadvantages of unsanctioned refinements*.

2.2.2.1.4.2 Unsanctioned use of "Concept Model attributes"

In some situations it may seem to be useful to use one of the attributes used in the *SNOMED CT Concept Model* to refine a *concept* that does not have a defining *relationship* or *qualifier* named by this attribute.

Provided that this is limited to qualifications that the *Concept Model* specifies for *concepts* of the same general type this approach can be applied. However, *Concept Model* attributes should not be applied to *concepts* of other types (for example the "approach"attribute should not be applied to a "finding"). Use of unsanctioned (but 'allowable') attributes for *refinement* may limit semantic interoperability.

Despite this limitation it may be appropriate to use a community agreed approach for a particular defined purposes. However, care should be taken to use attributes only in the manner described in the *Concept Model Guide*.

2.2.2.1.4.3 Use of "unapproved attributes"

The *SNOMED CT release* also includes a large number of attributes that are classified as "unapproved attributes".

Most of these originate from earlier terminology efforts. They have as yet not been applied in the *SNOMED CT Concept Model* and there is no guarantee that they will be used in a particular manner in the future.

This approach is not supported by the *SNOMED CT Concept Model*. Therefore any use of unapproved attributes for *refinement* is likely to limit semantic interoperability.

Despite these limitations, it may be appropriate to use a community agreed *Reference Set* of unapproved attributes within a defined user community for a particular defined purpose. Any such use should be fully documented by those responsible for its adoption. In the future as the *SNOMED CT Concept Model* evolves, additional supported attributes may provide a *migration* path for information recorded using a well-documented set of rules for a limited set of use cases.

2.2.2.1.4.4 Advantages and disadvantages of unsanctioned refinements

Note: THIS SECTION CONTAINS DISCUSSION NOTES ONLY.

The presence of defining or qualifying *relationships* certainly simplifies the task of implementing facilities for *refinement*. It also provides an indication that subsumption and *equivalence* computation may be possible. However, at this stage there is no definitive view of the extent to which *SNOMED CT* should sanction and permit particular *refinements* while deprecating or prohibiting other *refinements*.

Disadvantages of prohibition of all unsanctioned *refinements*

- **Lack of ability to express some required meanings**:
  - Until an attribute is included in the *Concept Model* and appropriately populated for all relevant *concepts*, it cannot be used to refine some *concepts* that might reasonably be so refined. The consequence of this are an inability to express some meanings required by users with approved *SNOMED CT expressions*.
- One example of this is that at present the following *expression* would not be sanctioned as headache has no associated severity *qualifier*. While this looks like an error that could readily be corrected it serves to illustrate the point.

25064002 | headache |:246112005 | severity |=24484000 | severe |

Disadvantage of allowing unconstrained *refinement*

- **Nonsense *expressions* with no "sensible" meaning**:
  - e.g. 25064002 | headache |:103366001 | with color |=414497003 | infra-red |
  - These are probably not a major cause for concern because it is impossible to create a foolproof approach that guarantees that all *expressions* will be sensible:
    - The following nonsense example is "sanctioned" in the sense that the site specified is a *refinement* of | head structure | which is the defined finding site for | headache |:
      - 25064002 | headache |: 363698007 | finding site |= 87056002 | infantile diploetic mastoid cell |

    - A nonsense *expression* is meaningless and where it is subsumed is largely irrelevant. Ideally it would subsume under nonsense *expressions* but that would require a knowledge of the rationality of all possible *expressions*.
    - In the absence of a tractable way to prohibit nonsense, avoidance and management of nonsense is an issue for implementers, users and qualify reviewers.

- **Nonsense *expressions* which may express a superficial "sensible" meaning**:

    - e.g. 25064002 | headache |:103366001 | with color |=301888000 | pale color |
    - A person reading this might think this expresses the fact the person's head (or face) was pale at the time of the headache. Logically in *SNOMED CT* it would mean that the headache is pale in color which is nonsense. However, an argument could be advanced that the same rules apply as those for indirect laterality and thus this could *transform* to:

        - 25064002 | aching pain |: 363698007 | finding site |=(69536005 | head structure |:103366001 | with color |=301888000 | pale color |).

    - This is still nonsense from a *SNOMED CT* perspective or perhaps it could correctly mean is | aching pain in the pale colored head structure |. However, if the author (or authoring application) assigned such an *expression* to represent two distinct findings | headache | and "head is pale in color " this meaning would not be apparent from a logical computational perspective.
    - While prohibition of nonsense is not tractable it may be feasible to state rules that express which forms of *expression* are logical and computable. Furthermore the outcome of these rules needs to be deterministic so that the result of transforming do not differ according to implementation.

- **Alternative rational *expressions* of similar meanings** :

    - Consider the following:

        1. 25064002 | headache |: 279114001 | character of pain |=410704005 | throbbing sensation quality |
        2. 162308004 | throbbing headache |
        3. *code*=162306000 | headache character | *value* =410704005 | throbbing sensation quality |

            - This assumes an information model with an observable entity *concept* naming a value in a separate information model attribute (*HL7* Observation supports this).

        4. 29695002 | throbbing pain |:363698007 | finding site |=69536005 | head structure |
        5. 25064002 | headache |:162306000 | headache character |=410704005 | throbbing sensation quality |.

    - All these *expressions* appear rational but only options 2 and 4 have the same *normal form* in the present *SNOMED CT Concept Model*.
    - Potentially option 3 could also be computed if both (a) the information model *terminology model* interface was clear and (b) the *SNOMED CT* definition of 162308004 | throbbing headache | is enhanced to add "363713009|has interpretation|=410704005|throbbing sensation quality|.
    - On the other hand option 1 is more in line with the way disorders are refined by "severity" and other qualitative *refinements*. For this to be computable equivalent the *concepts* "29695002|throbbing pain|" and 162308004 | throbbing headache | would both need revised definitions in which they were defined as having "279114001|character of pain|=410704005|throbbing sensation quality|".
    - Option 5 also looks superficially reasonable and shares the general feel of option 3. However, since 162306000 | headache character | is an "observable entity" rather than an "attribute" this representation would be contrary to one fundamental principle of *refinement* - that the name of the *refinement* should be a *subtype* of the *concept*"attribute". This means *current* normal *transform* rules would not result in a proper *normal form* and indeed might reasonable report an error.

- ***User interface* design issues** :

- Given all of the above points, application designers will struggle to create sensible and consistent interfaces unless advice on sanctioning is provided.
- Different issues will apply according to the nature of the interface. For example this may include:
  - What options to offer users to allow *refinement* of specific *concepts*;
  - How to represent the meaning that results from selecting options on a structured data entry form as a *SNOMED CT expression*;
  - How to encode meaning derived from *natural language processing*.

Interim recommendations

1. Wherever refining an existing defining or qualifying *relationship* enables representation of the required meaning this approach should be preferred.
2. Where 1 does not meet the requirements any attribute which is used in the *concept model* for *concepts* of the same type may be applied. The value applied to the attribute must be one of the allowable values as specified for that attribute in the *Concept Model Guide*:
   - For example a |causative agent| attribute can be applied to a clinical finding *concept*. The value assigned to this attribute is a value assigned from |Organism|, "physical force", "physical object" or |substance|. However, |causative agent| cannot be applied to refine a procedure. Furthermore the value of the |causative agent| cannot be a procedure or disorder.
3. Where neither 1 nor 2 meet the requirement use of additional attributes or values may be considered to meet a specific requirement. However, in this case, the implementer and/or user community will need to:
   - Avoid a direct conflict with other uses of the same attribute.
     - Ambiguity will arise if an existing attribute is overloaded to fulfill a different use-case:

     Example: The |laterality| attribute is used in the *concept model* to specify which of two functionally symmetrical paired structures is involved (e.g. "left wrist", "right kidney"). It should not be used for:

       - non-symmetrical structures (e.g. heart structures where the use of |left| and |right| refers to functionally different structures).
       - right or left side of a midline structure (e.g. |head|:|laterality|= |left| does not mean the "left side of the head" it means "left head" - and is thus not a useful *refinement*).
       - relative laterality (e.g. |trachea|:|laterality|= |left| does not mean "to the left of the trachea" or "trachea deviated to the left" it means "left trachea" - and is not a useful *refinement*).
   - Agree the approach to be taken in advance:
     - Ad-hoc *refinement* by end-users without any guidance on an agreed approach is liable to lead to multiple ways of representing the same required meaning and a loss of interoperability.
   - Document the approach taken in forms that:
     - Allow consistent use within the community;
     - Identify any issues related to computation of *equivalence* and subsumption between these local variant *expressions* and the content of *SNOMED CT*;
     - Are communicated to an appropriate *SNOMED CT* Working Group to help establish a wider consensus.
   - Make provision for future *migration* of data as a common *SNOMED CT* approach is developed in future.

Note: Within the UK, *NHS* Connecting for Health has issued guidance on post coordination which specifies *constraints* on allowable *refinements* and adds some specific *extensions* to the *refinements* sanctioned by released *relationships*. These guidance documents are available to implementers in the UK.

2.2.2.1.5 Applying values to concepts

Information model attributes such as values applied to an observable, also effectively refine the meaning of the *concept* as used in the record.

Currently the *SNOMED CT Concept Model* does not address issues of *equivalence* between a particular value applied to an observable or measurement procedure and a potentially similar finding (e.g. | creatinine measurement, serum | with a specified value and a finding such as | serum creatinine raised |).

There is a loose approximation using the | interprets | and | has interpretation | *Relationships* between some | clinical finding | *concepts* and relevant | observable entity | or | laboratory procedure | *concepts*.

Example: | serum creatinine raised | has a definition that includes:

- | interprets |=| creatinine measurement, serum |
- | has interpretation |=| above reference range |

Future enhancements: The *relationships* between | observable entity |, | laboratory procedure |, | evaluation procedure | and | clinical finding | *concepts* are currently under review.

2.2.2.2 Modeling semantic context

When a clinical finding is mentioned in a patient record certain assumptions are usually made about what it means in relation to the person who is the subject of that record. Thus if the finding | wheezing | is present in a record it is assumed to mean that the subject of that record is wheezing at the time of examination. This assumed meaning might be stated in full "the subject of the record is currently wheezing" but a contracted form that omits explicit reference to the subject, timing and presence of the finding is more usual in written records.

Similarly when a procedure is mentioned in a patient record assumptions are usually made about what it means in relation to the subject of that record. Thus, in the absence of other information, the mention of the procedure | cholecystectomy | may be assumed to mean that a "the subject of the record had a cholecystectomy at a stated time ".

Although default assumptions such as those above may be made, it is also possible for mention of the same finding or procedure to have a very different meaning. For example, "past medical history of wheezing", "not wheezing", "father suffers from wheezing", | cholecystectomy planned |, | cholecystectomy not done |.

The *SNOMED CT* context model provides a way to model *concepts* that explicitly state the *clinical situation* in which they are used. This same model also allows the construction of *expressions* that explicitly state the *clinical situation* in which a *concept* is being used in a particular record.

A proprietary record structure or a *reference information model* may also express aspects of context and these can be mapped to the *SNOMED CT* context model where appropriate to create comparable *expressions*.

The context model also specifies a default context that applies to findings and procedures which are expressed in a patient record without any explicit statement of context.

The most important aspects of the context model are those which have the potential to express a meaning that differs fundamentally from the meaning associated with the default context. Changes to context that have this fundamental effect on meaning are referred to as "axis modifying". The phrase "axis modifying" indicates a change that shifts the meaning between different axes in the *subtype hierarchy*.

The context model allows "axis modification" to be expressed within the general abstract logical model applied to all *SNOMED CT concepts*. To achieve this a *concept* such as | FH: Diabetes mellitus | is modeled as a *subtype* of | family history of disorder |. It is not a *subtype* of | diabetes mellitus | but instead its association with the finding | diabetes mellitus | is modeled using a defining *relationship* | associated finding |. Similarly a | Hip replacement planned | is a *subtype* of | planned procedure | (not a *subtype* of "hip replacement"). It is related to "hip replacement" by an | associated procedure |*relationship*.

2.2.2.3 Alternative logical abstract model views of expressions

Like a *concept*, an *expression* may be logically transformed into a variety of different views taking account of the definitions of the *concepts* which it references (i.e. the *Concept Identifiers* included in the *expression*).

2.2.2.3.1 Close-to-user expression view ("stated")

The close-to-user (or "stated") view of an *expression* contains references to the *concept* (or combination of *concepts*) together with *refinements* as selected by the user or as encoded by a clinical application to represent the semantics of a single clinical statement (i.e. a discrete clinical record entry).

The close-to-user view of an *expression* is the faithful representation of the information entered. For clinical safety and accountability purposes this should be regarded as the primary stored and communicated view of clinical information encoded using *SNOMED CT*.

Note: This view includes *refinements* applied by an application based on selections made in an entry form as well as those made explicitly. It does not include any *relationships* that are added based on *classifier* rules to make the *expression* complete or to normalize it.

2.2.2.3.2 Inferred expression views

An inferred *expression* can be derived from a stated *expression* by applying rules that take account of the definition of the refined *concept* and the associated refined values.

Inferences are drawn based on a consistent set of logical rules applied to the *expression* taking account of the definitions of *concepts* referenced by the *expression*.

Alternative semantically identical *expressions* may be generated using different logical *transformations*. The purpose of logical *transformations* is to support accurate and complete information retrieval through subsumption testing.

In general *terms* the types of *transformation* and resulting inferred views for *expression* are similar to those for *concept definitions*. The following sections of the guide identify some of inferred *expression* views and some of the differences between *expressions* and *concept definitions*.

2.2.2.3.3 Simple, nested and grouped expressions

A typical close-to-user *expression* consist of a single *concept* modified by optional *refinements* as shown in *Figure 18*. This may look like a *concept definition* but it is not defining the *concept* | hand pain |, it is specifying a more specific meaning by refining the | finding site | of the *concept* | hand pain | and adding a severity *qualifier*.

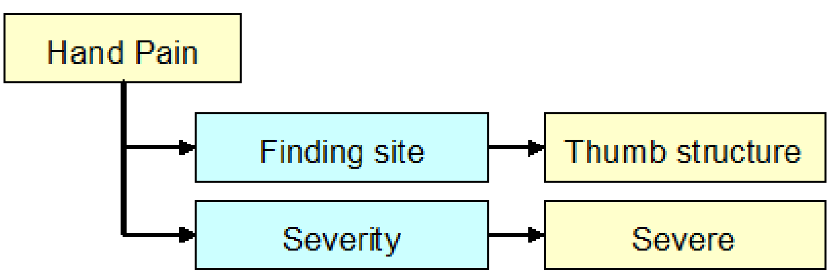


Figure 18: Refining a concept to add specificity

The target of a *refinement* may itself be refined producing a nested structure. An example of this is the application of laterality to finding site as shown in *Figure 19*.

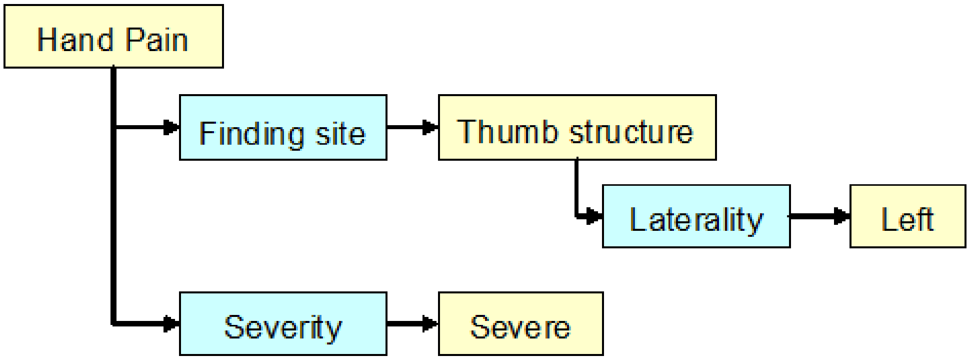


Figure 19: Nested refinement applied to a body site

In some cases, *refinements* within an *expression* may be grouped to represent association between a two different *refinements*. For example, a method and a target site or device as shown in *Figure 20*

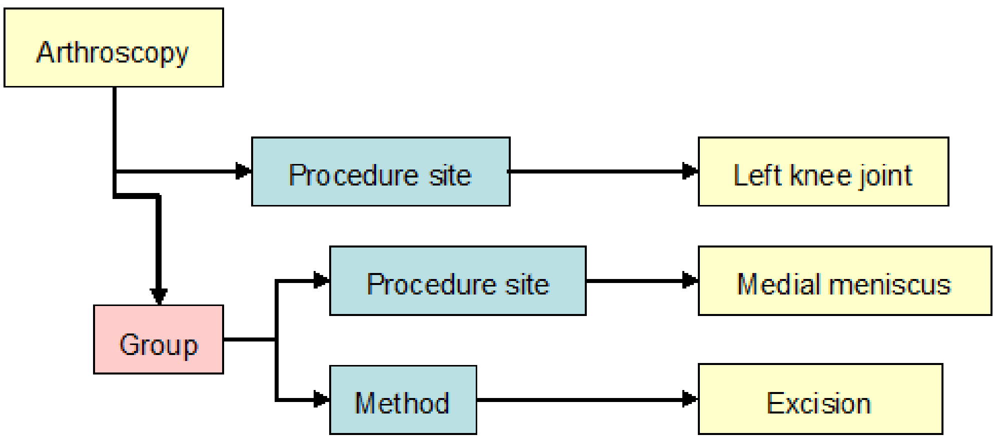


Figure 20: Grouped refinement

2.2.2.3.4 Expressions with multiple focus concepts

Some *expressions* may have multiple *concepts* followed by optional *refinements* as shown in *Figure 21*

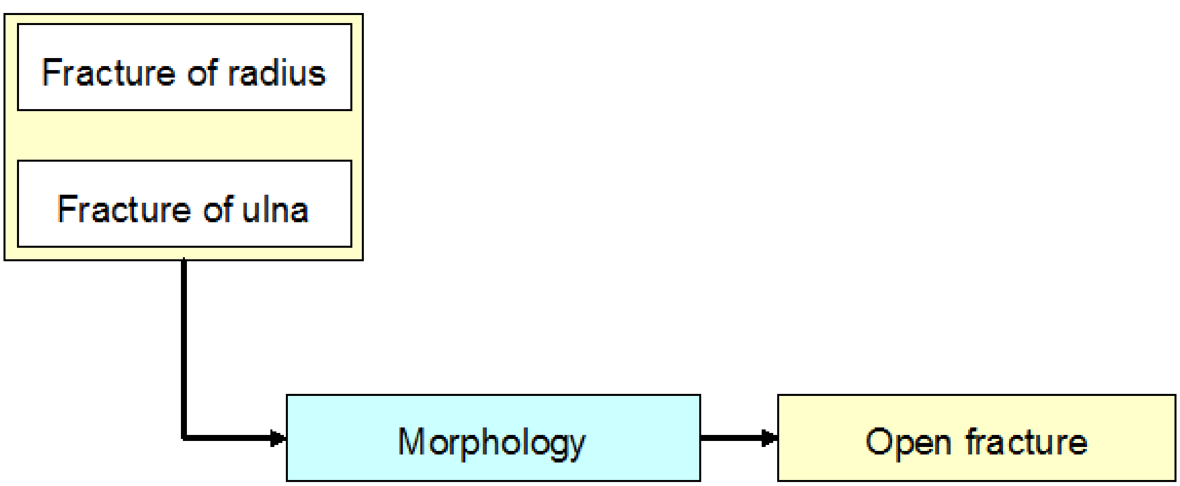


Figure 21: An expression with two focus concepts

The base of an *expression* may thus be one or more supertype *concepts* that are combined to produce a single meaning.

It is important to note that combining *concepts* at this level presumes that the result is intended to be a single combined meaning which is subsumed by the meaning of the combined *concepts*. Furthermore, the same *refinements* apply to the combined meaning of this set of *concepts*.

Some representational forms (e.g. *HL7* version 3 *Concept* Descriptor data type) do not allow combinations to be expressed in this way. However, it is possible to apply a simple logical *transformation* to create a semantically identical view that can be conveyed in a syntax that supports a single *focus concept* with *refinements* (see *Figure 22*).

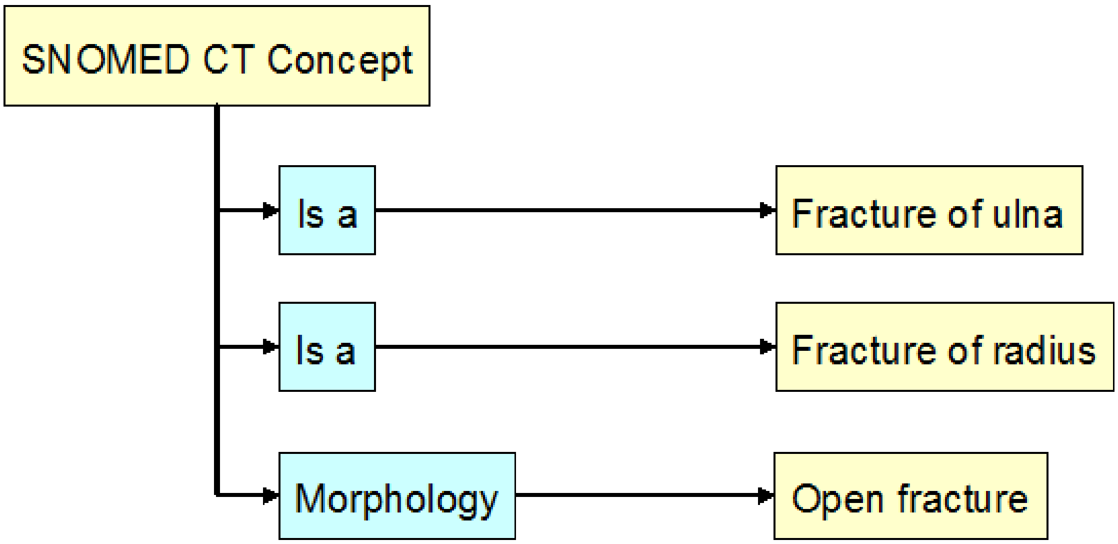


Figure 22: An alternative view of an expression with two focus concepts

2.2.2.3.5 Expressions that include context

Expressions may also explicitly represent the semantic context surrounding a finding or procedure. In these cases, the finding or procedure is nested inside the context component of the *expression*. The outer layer of the *expression*, which expresses the context, is sometimes referred to as the *context wrapper*. The nested *expression* representing the finding or procedure is sometimes referred to as the "clinical kernel".

Figure 23 illustrates how the general *concept* 281666001 | family history of disorder | can be refined to represent family history of a specific condition.

Figure 24 illustrates an alternative (computationally equivalent) representation of the same situation. In this case the family history situation is itself represented by an *expression*.

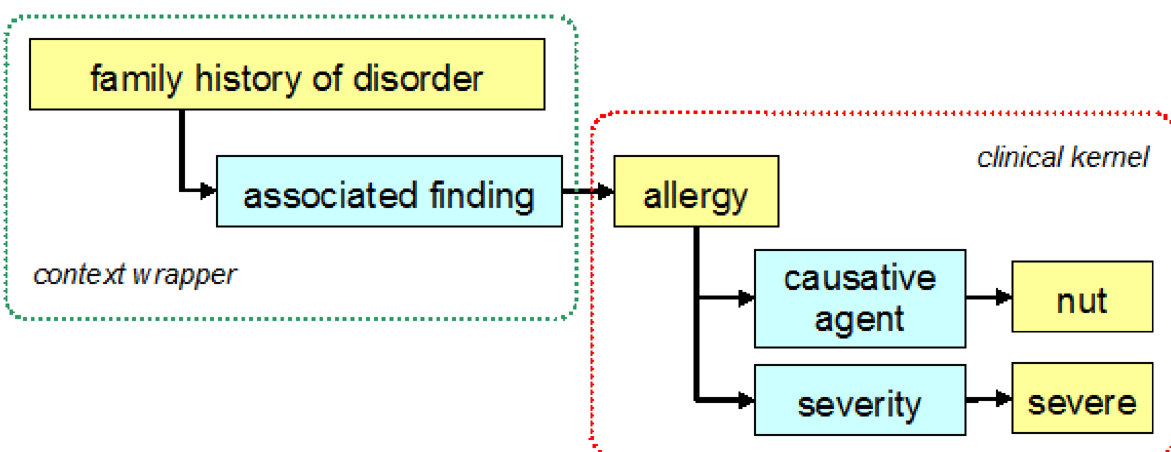


Figure 23: Family history of a specific type of severe allergy to nuts as close-to-user form expression

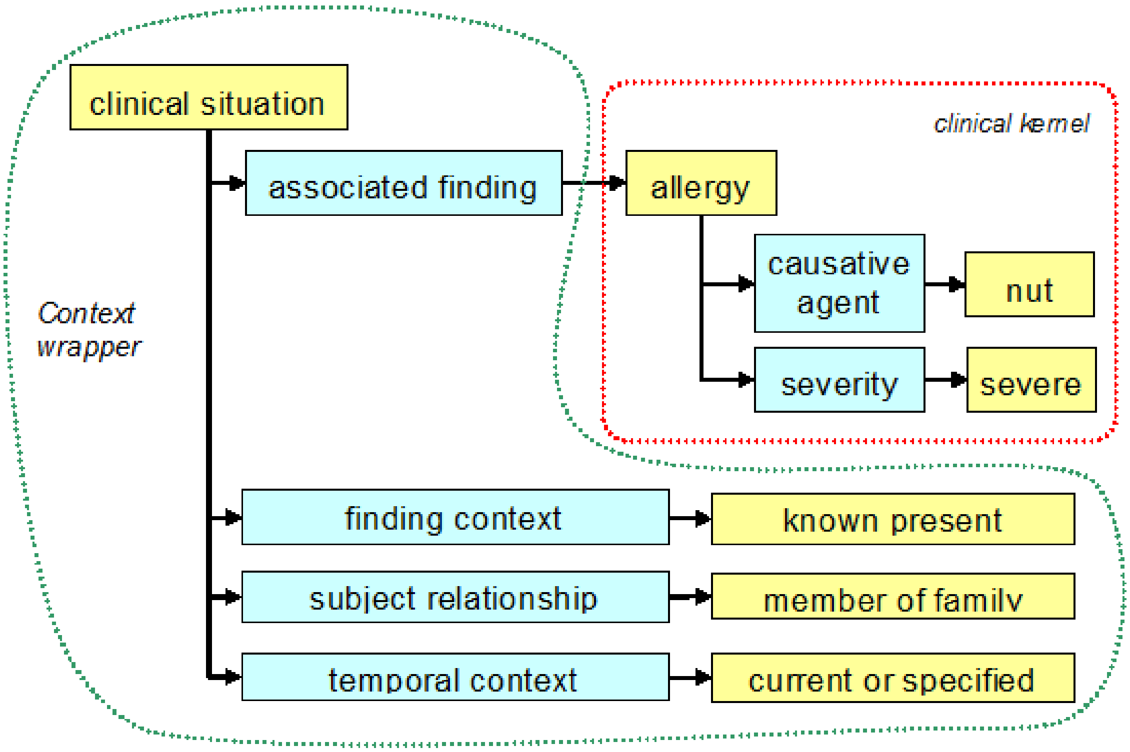


Figure 24: Family history of severe allergy to nuts represented by using a context wrapper expression

2.2.2.3.6 Normal form expression view

The theoretical range of equivalent *expressions* for a single idea includes two end-points:

- A fully *precoordinated expression* in which a single *concept identifier* is used to represent the idea;
- A maximally *postcoordinated expression* in which every facet of the idea is separately represented by an *attribute-value pair*.

In between these end points are a variable number of equivalent partially *postcoordinated expressions*.

Example: For a detailed example see *Example of equivalent postcoordinated expressions* on page 56

In order to compare *expressions*, is useful to be able to transform from these varied *expression* into a common *normal form expression*. This is possible using the combination of the *expression* and the definitions of the *concepts* to which it refers. A long as a reference *concept* is *fully defined* the defining *Relationships* for that *concept* can replace the *concept identifier* in the *expression*. This process reveals redundancies that can be removed by merging the definitions with the *expression*. An end-point is reached when all the *concepts* referenced by the *expression* are *primitive*. This is referred to as the *normal form*.

The process of normalization of *expressions* is described in detail in *Transforming expressions to normal forms*.

Note: The most important requirements for logical transformation of *expressions* is to enable information entered (in a close-to-user view) to be readily tested for *equivalence* or subsumption against another *expression* or to test inclusion within constrained range of values.

2.2.2.3.6.1 Example of equivalent postcoordinated expressions

To illustrate the range of possible equivalent *expressions* *Table 14* shows the *defining characteristics* of the hypothetical [3] *concept* "red steel pedal bike" and its *supertype ancestors*.

Table 14: Definitions of concepts used in illustration of alternative representations

Id	***Concept***	***Defining Characteristics***
1	Device (*PRIMITIVE*)	| is a | = Thing
2	Metal device	| is a | = Device Made of = Metal
3	Transport device	| is a | = Device Used as = Transport
4	Steel transport device	| is a | = Transport device | is a | = Metal device Made of = Steel Used as = Transport
5	Pedal powered transport device	| is a | = Transport device Used as = Transport Used as = Transport Power = Pedals
6	Bicycle (*PRIMITIVE*)	| is a | = Transport device Used as = Transport Moves on = 2 wheels
7	Pedal bicycle	| is a | = Pedal powered transport device | is a | = Bicycle Used as = Transport Moves on = 2 wheels Power = Pedals

[3] This hypothetical concept is chosen in preference to a real SNOMED CT concept to allow illustration of theoretical points with simple qualifiers. While all the points illustrated apply to some SNOMED CT concepts but there is no single concept that readily illustrates all these points without introducing other issues or having a long name that complicates the illustration.

Id	*Concept*	*Defining Characteristics*
8	Red pedal bicycle	| is a | =Pedal bicycle | is a | =Pedal powered transport device Used as = Transport Moves on = 2 wheels Power = Pedals Color = Red
9	Steel pedal bicycle	| is a | =Pedal bicycle | is a | =Steel transport device Used as = Transport Moves on = 2 wheels Power = Pedals Made of = Steel
10	Red steel pedal bike	| is a | = **Red pedal bicycle** | is a | = **Steel pedal bicycle** Used as = Transport Moves on = 2 wheels Power = Pedals Made of = Steel Color = Red

Figure 25 illustrates a range of *expressions* based on each of the *concepts* defined in might be used to represent the *concept* "red steel pedal bike".

Expression K is a *precoordinated expression* using the *concept* "10 | red steel bike| ". Each of the other forms is *postcoordinated* by adding refinements that build on the *concept* definitions shown in *Table 14*.

These *expressions* would all be equivalent if the definitions were complete and accurate. In that case, it would possible to transform between them without losing information by appropriately adjusting the associated refinements to take account of the *concept* definitions. In practice the *concept* "bicycle" is marked as *primitive* which places a limit on transformation process.

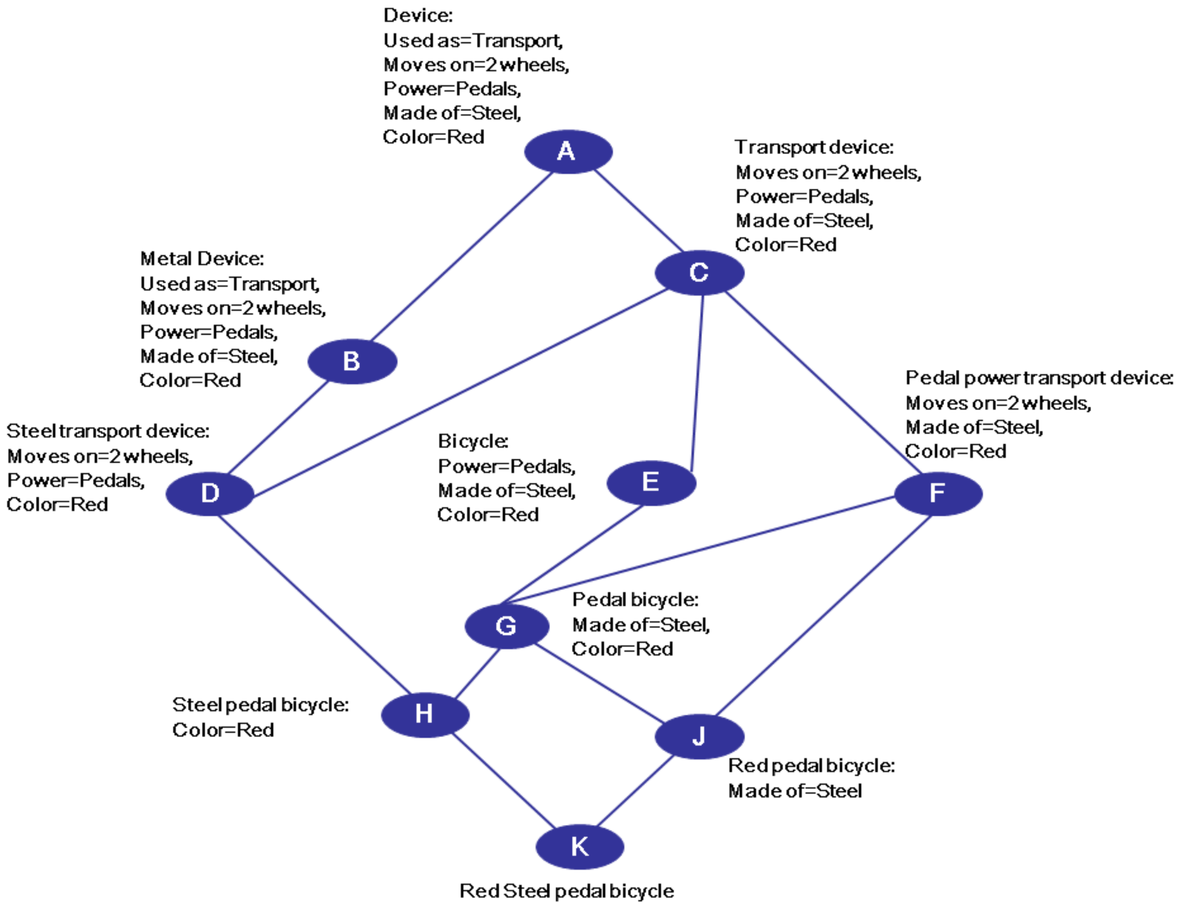


Figure 25: Alternative expressions that mean "red steel pedal bicycle"

Two rules limit the range of equivalent *expressions*:

Rule 1: It is not possible to transform a *primitive concept* into a *postcoordinated expression*.

- A *primitive concept* has a facet that is not represented by its *defining characteristics* and therefore any attempt to represent it in a *postcoordinated* form results in a loss of information.

This is illustrated by consideration of the definitions of the *concept* "bicycle" in *Table 14*. The definition stated in the table is as follows:

| is a | " = "Transport device", | Used as | = | Transport |, | Moves on | = | 2 wheels |, | Origin | = | Man made |

This definition would also apply to a horse-drawn cart or a trailer. Therefore the *concept* "bicycle" must be regarded as *primitive*. Recognizing this fact means that some of the apparently equivalent *expressions* in *Figure 25* cannot be computed as equivalent. Unless the *focus concept* is a *subtype* of "bicycle" it is not possible to compute that it is a kind of bicycle. This means that to create an equivalent *expression* it would be necessary to add | is a | = "Bicycle". This is shown in *Figure 26*.

Examining these definitions, it is apparent that the characteristics shown in gray are redundant because they are part of the definition of "bicycle."

As a consequence of this rule, *primitive concepts* create the limits on the ability to transform an *expression* to a more post-coodinated form. An *expression* can be normalized until all the *concepts* referred to by the *expression* are *primitive*. An *expression* in which all the referenced *concept* are *primitive* is referred to as the *normal form*.

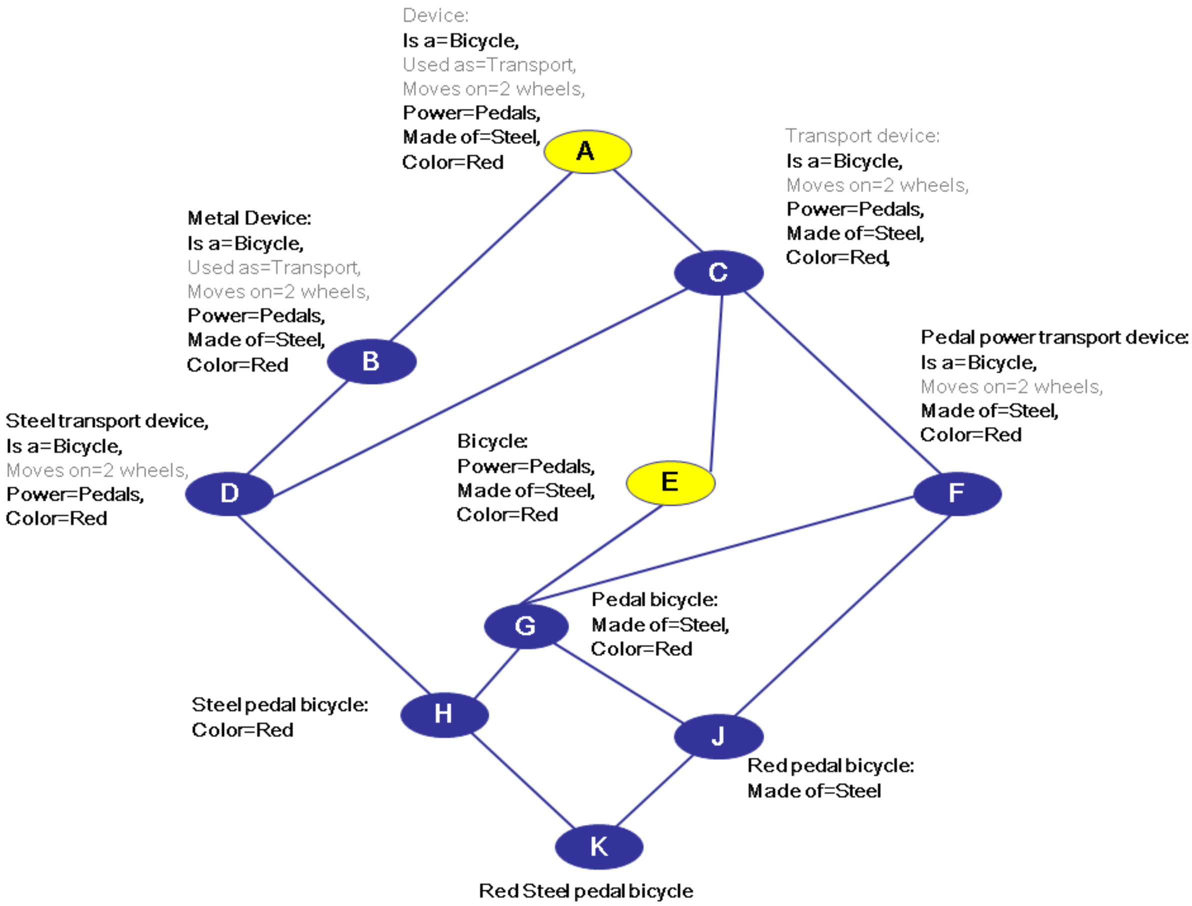


Figure 26: Expressions meaning "red steel pedal bicycle" with "bicycle"recognized as primitive

Rule 2: It is not possible to transform a *postcoordinated expression* into a fully *precoordinated concept* unless such a *concept* already exists in the released terminology.

This second rule is perhaps self-evident but it is stated because, like the first rule, it alters the available representations. If the *concept* "red steel pedal bicycle" was not available in a *precoordinated* form, there are two distinct *expressions* that are as *precoordinated* as possible (i.e. "steel pedal bicycle" + "color " = "red" and "red pedal bicycle" + | made of | = | steel |). This is illustrated in *Figure 27*. In such cases there is no obvious reason to prefer one compared to the other.

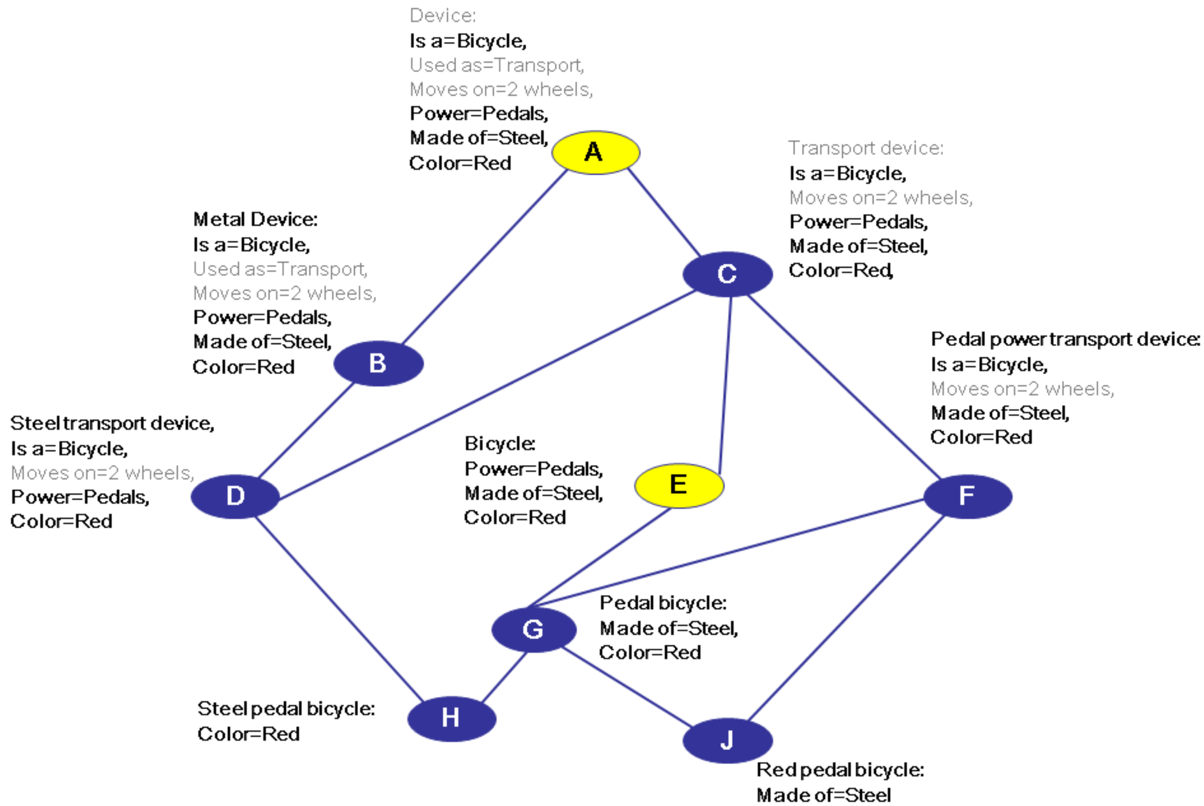


Figure 27: Possible expressions of "red steel pedal bicycle" without precoordinated option

2.3 Representational Forms

This section describes different ways in which *SNOMED CT components*, *derivatives* and *expression* can be represented. These representations include the files in which *SNOMED CT* is distributed as well as possible representations that may be used assist implementation or optimize particular functions.

2.3.1 Release Files

SNOMED CT is provided to licensees as a set of *release files*. The file naming conventions and the structure of these files is described in *Release File Specifications* in a separate section of this guide. There are currently two distinct *Release Formats*:

- *Release Format 1* (*RF1*): The specification in which *SNOMED CT* has been provided since its first release in 2002 (with a few minor amendments).
- *Release Format 2* (*RF2*): Based on a draft trial specification that adds a range of significant enhancements.

2.3.2 Representing SNOMED CT identifiers

Components within *SNOMED Clinical Terms* 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 *Identifier* 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 *Identifier* also supports separate *namespaces* that distinguish between different issuing organizations .

2.3.2.1 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".

2.3.2.2 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 in that form of the identifer. 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 *IHTSDO* 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 *Extensions* created by an appropriately authorized organization.

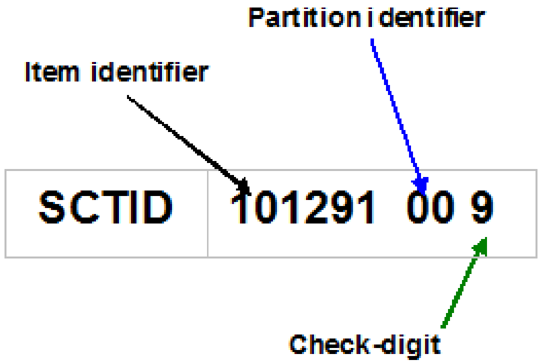


Figure 28: SCTID Short Format - Applicable to components originating from the International Release

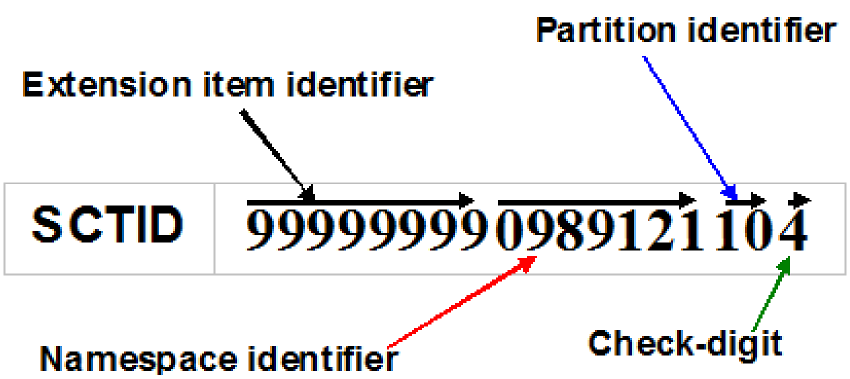


Figure 29: SCTID Long Format - Applicable to components originating from a SNOMED CT Extension

Note: The *IHTSDO* allocates *namespace-identifiers* to organizations such as *IHTSDO Members* and *Affiliates* to enable them to create content and or derivatives in an *Extension*. The *namespace-identifiers* enables unique *SCTIDs* to be issued by many organizations and allow each *SCTID* to be traced to an authorized originating organization .

2.3.2.3 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 *strings* 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*.

Notes:

1. 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.
2. 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).
3. *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 sets* can be used to find the identifier of the closest equivalent *active concept*.

2.3.2.4 Check-digit

The final (units) digit of the *SCTID* is the *check-digit*. It is not envisaged that users will be routinely required to type *SCTID* values. However, the objective of the *check-digit* is to detect the commonest types of error that may occur due to typographical errors on those occasions 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.

See *Check-digit computation* for detailed information about the Verhoeff *check-digit* and sample program code.

2.3.2.5 Partition-identifier (RF1)

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:

00	A Concept
01	A Description
02	A Relationship
03	A Subset
04	A Cross Map Set
05	A Cross Map Target

Identifiers of components in Extensions have one of the following partition-identifier values:

10	A Concept in an Extension
11	A Description in an Extension
12	A Relationship in an Extension

13	A Subset in an Extension
14	A Cross Map Set in an Extension
15	A Cross Map Target in an Extension

All other partition-identifier values are reserved for future use.

2.3.2.6 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 *SCTIDs* is allocated a *namespace-identifier* by the *IHTSDO*. 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*).

2.3.2.7 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 *item-identifier* on it is meaningless. The only way to determine the meaning of an *SCTID* is by looking up the complete value in an appropriate distribution file.

2.3.2.8 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*.

SctId	*Partition identifier*	*Check digit*	Notes
1290023401004	00=*Concept*, using short format	9	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*.
10000001105	10=*Concept*, using long format	5	A valid long format SctId for a *Concept* in the 0000001 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.

The Namespace hierarchy

SNOMED CT core release files include metadata *Concepts* that represent each of the allocated *namespace-identifiers*. The *Concepts* representing the namespaces are arranged in a single parent hierarchy, as follows:

- 370136006 | Namespace concept |
  - 373872000 | Core Namespace |
    - | Extension Namespace A {} |
    - | Extension Namespace B {} |
      - | Extension Namespace D {} |
      - | Extension Namespace E {} |

- | Extension Namespace C {} |

Figure 30: Hierarchy for: Namespace concept (namespace concept)

In the above hierarchy,| Extension Namespace A {} |, | Extension Namespace B {} | and | Extension Namespace C {} | are all child namespaces of the 373872000 | Core Namespace | (representing the *International edition* which does not have a *namespace-identifier*, and uses short format *SCTIDs* to identify *components*). Also, | Extension Namespace B {} | is the parent namespace of | Extension Namespace D {} | and | Extension Namespace E {} | .

Each *Namespace concept* may only have one parent *Namespace concept* in the 370136006 | Namespace concept | sub-hierarchy.

The namespace hierarchy is used to constrain which content can be promoted from one *Extension* to another without amending the *SCTID*. Content may be moved (without amendment of *SCTID*) from an *Extension* released by the owner of a child namespace to an *Extension* released by the owner of a parent (or ancestor) namespace, as described by the |370136006 | Namespace concept | sub-hierarchy.

Examples:

1. A *concept* with an *SCTID* that includes | Extension Namespace D {} | may be moved to the *Extension* maintained by the owner of | Extension Namespace B {} | without changing its *SCTID*, because this is a parent of the originating namespace.
2. A *concept* with an *SCTID* that includes | Extension Namespace D {} | must not be moved to the *Extension* maintained by the owner of | Extension Namespace C {} | because this is not parent (or ancestor) of the originating namespace. Therefore, to make this move the original *concept* must be inactivated and replaced by a new component with a new *SCTID* in target namespace.
3. Any *concept* may be moved from any *Extension* to the *International Release* (subject only to formal acceptance that is a valid addition for international use).

Namespace concepts have the following characteristics:

- They are *subtypes* (either children or *descendants*) of 370136006 | Namespace concept |.
- The *Fully Specified Name* of each *Concept* has the form " *Extension* Namespace {nnnnnnn} (*namespace concept*)" – where nnnnnnn is the seven digit Namespace-*Identifier*.
- A *Synonym* associated with each *Concept* has the form " *Extension* Namespace nnnnnnn"
- Where appropriate further *Synonyms* may be included to identify the nature of the responsible organization.

When requesting a *namespace-identifier* from *IHTSDO*, there will be a facility to optionally specify a parent *Namespace-identifier* for the new namespace.

To specify a parent namespace for an existing *namespace-identifier*, please contact *info@ihtsdo.org* with details of your existing *namespace-identifier* and its proposed parent *namespace-identifier*.

Caution: Once a *namespace-identifier* has be allocated a parent *namespace-identifier* in this hierarchy, further changes to this hierarchical *Relationship* are not permitted. This restriction is imposed to avoid changes that would undermine traceability of moves between namespaces.

2.3.3 Representing Extensions

2.3.3.1 Extension Tables - Structure

Extensions use the same table structure as the *Concepts*, *Descriptions*, *Relationships*, and *Reference Sets* tables defined in those respective sections of this manual. These tables have the same structure or schema as the *core tables* but are in separate files.

When packaged, *extension* file names should follow the conventions defined by the *IHTSDO*. For more information, refer to the document *SNOMED CT File Naming Convention*.

2.3.3.2 Specification for Namespace within the SCTID

The *identifiers* assigned to all *components* that originated as part of an *extension* include a *namespace-identifier* (see *Representing SNOMED CT Identifiers*). This means that the sets of *Identifiers* available to each organization authorized to issue components are distinct, which ensures that the same *Identifier* cannot be issued by two different organizations.

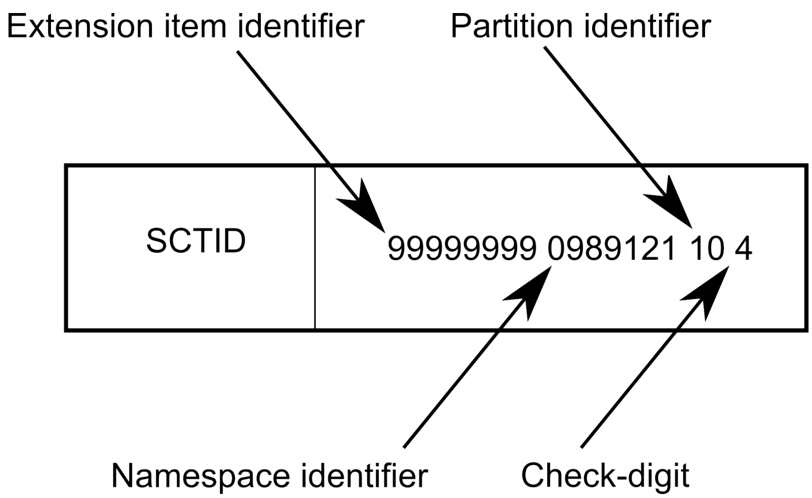


Figure 31: SCTID for a concept originated as part of an Extension

All *Extension components* (rows) originated by an organization must use the *Namespace Identifier* assigned to that organization . *Namespace Identifiers* are issued by the *IHTSDO* so that the *Namespaces* remain unique between organizations . Allocated *Namespaces* are recorded as *Concepts* in the |SNOMED CT Model Component |*hierarchy* ("special *concepts*" hierarchy in *RF1*) when they are issued to an organization .

Namespaces serve three *Roles*:

- Preventing collision or reuse of *SNOMED CT identifiers*;
- Indicating the origin of a component. In RF2 responsibility for maintenance is tracked only by the ModuleId, this is the field that should be used to avoid potential risk of two organizations make conflicting changes to the same *component*.
- Indicating the source for information about a *concept* - relevant for *Extension Concepts* that are not directly available in a particular system.

All *Extension components* (rows) should use the appropriate *partition-identifiers* for *Extensions*. This ensures that *components* of the *SNOMED CT International Release* can be distinguished from *components* that are part of an *Extension*.

Note: *Components* that originate as part of the *International Release* do not have a *namespace* field and are distinguished instead by *partition-identifier* values that are specific to *International Release*.

Note: Each organization can assign the item-*Identifier* portion of the *SCTID* in any way within its *Namespace*. If there is a need to allocate part of the development process to a subdivision within an organization , they may be allocated a set or range of *item-identifiers* that have not yet be used or allocated within that *Namespace*. The authorized organization must ensure that it tracks and manages all such allocations in a way that avoids any risk of reuse of the same *SCTID*.

2.3.3.3 Namespace allocation

*Namespace-identifier*s are allocated by the *IHTSDO* to licensed organizations . The *IHTSDO* is under no obligation to allocate a *namespace* to any organization and makes these allocations at its discretion.

Allocation of a *namespace* does not imply any endorsement of the reputation of an organization nor to the quality or fitness for purpose of any *Extensions* created by that organization . Users and/or vendors incorporating *Extensions* into their application do so at their own risk and should satisfy themselves with the reputation of the responsible organization and the quality the *Extensions* so incorporated.

2.3.3.3.1 Namespace Allocation Policy/Regulation

Title: *Namespace* Allocation Policy/Regulation

Effective Date: August 4, 2009 **Owner** : Management Board

Date Last Reviewed: October 14, 2009 **Date Last Revised** : October 14, 2009

2.3.3.3.1.1 Regulation Statement

IHTSDO will allocate *SNOMED CT Namespace Identifiers* upon written request from a Member or an Affiliate in accordance with the procedures outlined below. The *IHTSDO* will also maintain and publish a register of *Namespace Identifiers* issued.

Section 9 of the Articles of Association provides the starting point for the *Namespace* Allocation Policy. It states that:

- 9.1 Only the Association may issue *Namespace Identifiers*.
- 9.2 The Association shall, upon written request from a Member or an Affiliate in accordance with such procedures as the Association may prescribe by Regulations, issue one or more *Namespace Identifiers* to the Member or Affiliate. The Association shall not unreasonably refuse to issue a *Namespace Identifier* to a Member or an Affiliate.
- 9.3 The Association shall be responsible for ensuring that each *Namespace Identifier* is only issued to a single Member or Affiliate.

In addition, section 7.1.7 states that "An Affiliate may not create any Standards-Based Third Party *Extension* or any Standards-Based *Derivative* from the Member's National *Extensions* unless that Affiliate has been issued with a *Namespace Identifier*."

2.3.3.3.1.2 Definitions

Affiliate: An Affiliate of *IHTSDO* in accordance with *IHTSDO*'s *Articles of Association*, i.e. a person or organization to which the *International Release* of *SNOMED CT* (whether on its own or as part of a Member's *National Release* of *SNOMED CT*) is distributed or otherwise made available under the *Articles of Association.*

Namespace Identifier: A code or that part of a code that identifies the organization responsible for creating and maintaining a standards-based *extension* or a standards-based *derivative*. It is an element of *SNOMED CT concept Identifiers*.

2.3.3.3.1.3 Context

Namespace Identifiers are 7-digit numbers that *IHTSDO* issues to those who create *extensions* to *SNOMED CT*, such as national *extensions*. *Namespace identifiers* ensure that it is clear who developed and maintains particular customized terminology. They also ensure that terminology in *SNOMED CT extensions* has unique *Identifiers* but a common structure, which facilitates application development and the creation of *Reference Sets*. There is a defined process for management of *Namespace Identifiers* when terminology is moved between *extensions* or from an *extension* into the *International Release*.

It should be noted that this policy covers the technical mechanism to allocate *Namespace Identifiers* in order to be able to identify the source of content, prevent collisions in terminology that would affect interoperability, and achieve similar goals. It does not cover whether or not particular types of content, including *extensions* and *derivatives*, can be used in a given context. This may be the subject of national policies, guidelines, or other documents. Requesters of a *Namespace Identifier* are encouraged to review *SNOMED CT International Release* documentation and to consult with *IHTSDO Members* in countries in which deployment of any content developed in the *Namespace* is planned for additional guidance, policy, and/or process documents which may be relevant.

2.3.3.3.1.4 Procedures

2.3.3.3.1.4.1 Informing the Community of Practice :

IHTSDO will inform the Community of Practice about the process for requesting a *Namespace Identifier*.

- A copy of this regulation will be posted to the *IHTSDO* website.
- Instructions for requesting a *Namespace Identifier*, including the form for making such a request, will be posted to an appropriate location on the *IHTSDO* website (e.g. the Frequently Asked Question or "How do I?" pages).
- Confirm which Members would like to be notified when one of their Affiliates requests a *Namespace*, i.e. an Affiliate listing an address in the jurisdiction in question on their *Namespace Identifier* Application Form and/or who identified that they received their Affiliate License through that jurisdiction.

2.3.3.3.1.4.2 Requesting & Granting a Namespace:

The Association shall, upon written request from a Member or an Affiliate in accordance with such procedures as the Association may prescribe by Regulations, issue one or more *Namespace Identifiers* to the Member or Affiliate. The Association shall not unreasonably refuse to issue a *Namespace Identifier* to a Member or an Affiliate.

- To request a *Namespace Identifier*, individuals/ organizations should complete and submit a copy of the *Namespace Identifier* Application Form by email to support@ihtsdo.org.
- *IHTSDO* should verify that the requester is either:
  - An *IHTSDO Member*
  - An individual or organization who holds a valid AffiliateLicense
  - An individual or organization who does not fall into the above categories but whose application is approved in writing by the *IHTSDO*
- If the conditions above apply, *IHTSDO* should issue a unique *Namespace Identifier* to the requester if:
  - The request is from an *IHTSDO Member*;
  - The request is from the holder of a valid Affiliate License , is for a single *Namespace Identifier*, and the requester has not already been issued a *Namespace Identifier*; OR;
  - *IHTSDO*'s CEO approves the request in writing.
- The issuance of the *Namespace Identifier* should be confirmed in writing with the requester, along with a link to this policy and a reminder that they will be contacted annually to reconfirm their contact information and potentially provide additional information to be published in the register. Requesters should also be provided with Member contact information and the recommendation that they should contact relevant Members to obtain any additional national guidance, policy, and process documents which may be relevant.
- The relevant Member should be informed if they have requested to be notified when an Affiliate from their jurisdiction requests a *Namespace*, i.e. an Affiliate listing an address in the jurisdiction in question on their *Namespace Identifier* Application Form and/or who identified that they received their Affiliate License through that jurisdiction.

2.3.3.3.1.4.3 Format of Namespace Identifiers:

- *Namespace Identifiers* are 7 digit numeric codes;
- *Namespace Identifiers* are issued in sequence, unique, and not re-used.

2.3.3.3.1.4.4 Maintaining and Publishing a Namespace Register:

IHTSDO will maintain and publish an up-to-date *Namespace* Register.

- When *Namespace Identifiers* are allocated, a record of the number of that *Identifier*, the date of issuance, the body from which the Affiliate License was obtained (if applicable), and the name and contact information of the individual/organization to which it was issued will be added to the *Namespace* Register.

- On an annual basis, *IHTSDO* will contact all those to whom *Namespace Identifiers* have been issued by email to confirm contact information. A reminder will be sent after 30 days if a response has not been received. The year in which a confirmation of current contact information was last received will also appear in the *Namespace* Register.
- *IHTSDO* reserves the right to make a *Namespace Identifier inactive* for current and future use (i.e. it cannot be used for newly-created *concepts* from that point onward) if the individual or organization to which it was issued cannot be contacted after three attempts. This *status* will be noted in the *Namespace* Register and the individual or organization to which the *Namespace Identifier* was issued will be notified accordingly.
- *IHTSDO* also reserves the right to make a *Namespace Identifier inactive* if (1) it is requested to do so by the organization to which the *Namespace Identifier* was issued, (2) the organization to which the *Namespace Identifier* was issued is involved in a merger or acquisition with another organization to which a *Namespace Identifier* has been issued, or (3) it receives a written complaint about the use of that *Namespace* that, upon investigation, it determines to be well-founded, according to the protocol for material breaches and termination of Affiliate Licenses identified in clause 5.2 of the Affiliate License .
- The *Namespace* Register will be published with each version of the *SNOMED CT International Release*. In the future, *IHTSDO* reserves the right to also publish the NamespaceRegister on the *IHTSDO* website.

2.3.3.3.1.5 References

- *IHTSDO Articles of Association* ;
- *SNOMED CT Technical Implementation Guide* ;
- *IHTSDO Namespace Identifier Application Form* .

2.3.3.3.1.6 Document Control

This policy was approved by the *IHTSDO* Management Board on August 4, 2009 and is subject to regular review according to *IHTSDO*'s policy review processes. Key stakeholders include the Technical Committee, the Implementation and Innovation Committee, the Member Forum, and the Affiliate Forum.

2.3.3.4 Component Guidelines

Descriptions that are part of an *Extension* can refer to either a *Concept* that is part of that *Extension*, a *Concept* that is part of another *Extension*, or an *International Release Concept*.

Relationships that are part of an *Extension* can relate two *Concepts* in the *Extension* or two *Concepts* in different *Extensions*. The *relationship* can also relate the *extension Concept* to an *International Release Concept*-- that is, *sourceId* is in the *extension* and *destinationId* is in the *International Release*.

2.3.3.5 Content of Extensions

The *components* in an *Extension* have the same structure as *International Release components* of *SNOMED CT*.

SNOMED CT International Release is not dependent on availability of any *Extension*. However, all *SNOMED CT Extensions* are dependent on the *SNOMED CT International Release*. Some *Extensions* may also be dependent on other *Extensions*. Dependencies between *Extensions* must be declared and must not be circular.

The following rules apply to dependencies between *components* and *derivatives* in *Extensions*.

Note: In these rules,

- *Containing-Extension*, refers to the *Extension* that contains the named *component* or *derivative*.
- *Dependee-Extension*, refers to another *Extension* on which the *Containing-Extension* is dependent.

1. Every *Concept* in an *Extension*:

  - Must be a *subtype descendant* of an *International Release Concept*:

    - This descent may be indirect, passing through *Concepts* in either the *Containing-Extension* or a *Dependee-Extension*.

2. Every *Description* in an *Extension*:
    - Must apply to a *Concept*, in the one of the following *Namespaces*: the *Containing-Extension*, the *International Release* or a *Dependee-Extension*.
3. Every defining *Relationship* in an *Extension*:
    - Must define a *sourceId* which refers to a *Concept* in the *Containing-Extension*.
        - In exceptional circumstances, an *Extension* may add additional defining attributes to a *Concept* in the *International Release* or a *Dependee-Extension*.
    - Must have *typeId* which refers to a *Concept*, in the one of the following *Namespaces*: the *Containing-Extension*, the *International Release*, or a *Dependee-Extension*.
    - Must have a *destinationId* which refers to a *Concept*, in the one of the following *Namespaces*: the *Containing-Extension*, the *International Release*, or a *Dependee-Extension*.
4. A *Reference Set* in an *Extension*:
    - May include references to *components* and *derivatives* in any of the following *Namespaces*: the *Containing-Extension*, the *International Release*, or a *Dependee-Extension*.
5. The enumerated values used in *RF2 components* that form part of an *Extension* must be all be represented by metadata *Concepts* that are present in one of the following *Namespaces*: the *Containing-Extension*, the *International Release*, or a *Dependee-Extension*.

The following additional rules are only relevant to *Extensions* represented using *Release Format 1*:

- *Reference Sets* in an *Extension*:
    - May include as its members *Components* from the following *Namespaces*: the *Containing-Extension*, the *International Release*, or a *Dependee-Extension*.
    - May refer to other *Reference Sets* in the following *Namespaces*: the *Containing-Extension*, the *International Release*, or a *Dependee-Extension*.
    - May provide maps for *Concepts* from the following *Namespaces*: the *Containing-Extension*, the *International Release*, or a *Dependee-Extension*.

2.3.3.6 Transfer of Responsibility between Organizations

When the need arises to transfer *components* (*Concepts*, *Descriptions*, *Relationships*) from the *International Release* content to an *Extension*, from an *Extension* to the *International Release*, or from one *Extension* to another, conversation and coordination between the sending and receiving organizations is needed. In some cases, entire tables may be transferred - not just individual components.

It should be noted that the transfer of *components* among *Extensions*, or between an *Extension* and the *International Release*, is subject to the terms of the *IHTSDO Affiliate* License and, within an *IHTSDO Member* country, may also be subject to the *terms* of that Member's *SNOMED CT* national license .

Examples of transfers include:

- From the *SNOMED CT International Release* to an organization responsible for an *Extension*:
    - This occurs if a decision is made that some *Concepts* in the *International Release* are specific to a *Realm* or domain or interest for which another organization has been allocated responsibility:
        - For example, this applies to UK specific drugs and UK specific administrative *Concepts* which are maintained by the *UK NHS*.

- From an organization responsible for an *Extension* to the *International Release*:
  - This occurs if an organization recognizes that some of its *Extension* content belongs in the *International Release* as it has general applicability;
  - It also occurs if an organization hands over responsibility for its entire *Extension* to the *IHTSDO*.
- From one organization responsible for an *Extension* to another organization :
  - This occurs if one organization recognizes that some of its *Extension* content belongs in a domain managed by another organization ;
  - It also occurs if an organization hands over responsibility for its entire *Extension* to another organization .

There are three types of transfer of responsibility:

- Transfer of an entire *Extension* (i.e. all *components* ever issued with an *SCTID* in a given *Namespace*. from one organization to another organization):
  - This is a straight forward process. All that happens is that another organization assumes responsibility for the original *Namespace-identifier*. There is no need for detailed tracking of individual *components*.
- Transfer of one or more *components* from an *Extension* to the *International Release* or to a "parent"*Extension*:
  - As a result of revision to guidance issued by the *IHTSDO* in 2011, these transfers can be made without changing the *Identifier* of the *component* provided that the RF2 format *moduleId* field is used to denote that the *component* is now being issued as part of a different module.
- Transfer of one or more *components* between other *Extensions* or from the *International Release* to any *Extension*:
  - In this case, the *Namespace* is not transferred and thus, to fulfill the roles of the *Namespace-Identifier*, the *component* must be assigned new *SCTIDs* in the *Namespace* of the newly responsible organization :
    - The previous instances of these *components* are withdrawn from *current* use with the *Status* value *Moved Elsewhere*;
    - Appropriate *Relationships* point to replacement *components* in the new *Namespace*.

The transfer of responsibility depends on the release schedules of the organizations involved. Often the original source organization will be aware of an intended move before the target organization has accepted responsibility and released the *component*. To facilitate this, an interim *Status* value *Pending Move* is applied to *components* that are being moved to another *Namespace* but are intended for *active* use until their replacements are found in the target *Namespace*.

To provide continuity for a *Concept* if responsibility is transferred, the *Concept Status* and history are coordinated as follows:

Table 15: Processing Transfers between Extensions

	Sending Organization	Receiving Organization [4]
Start State	*Concept* A *Concept Status* = *Current*	

[4] **Assume assigned namespace = 9999999**

	Sending Organization	**Receiving Organization** [4]
Agreement to transfer responsibility	*Concept* A *Concept Status* = Move Pending	
Responsibility Transferred	*Concept* A *Concept Status* = *Moved Elsewhere* History Files = *Concept* A "*Status* Change" to "*Moved Elsewhere*" History Files = *Concept* A "Moved to" *Namespace* 9999999 in this release **Note:** *Namespace* 9999999 is recorded as a "Special *Concept* " in the *SNOMED CT Concepts* File. Therefore, the Sending Organization can track the organization to which the *concept* has moved, even if the new *Concept Identifier* is not yet assigned.	*Concept* B (*Concept* B is a new *Concept Identifier* using *namespace* = 9999999) *Concept Status* = *Current* History Files = *Concept* B "Added" History Files = *Concept* B "Moved from" *Concept* A **Note:** The Receiving Organization can record the *Concept Identifier* previously used for the *concept*. **Note:** If the Receiving Organization is the *IHTSDO*, the *namespace* would be the *International Release namespace* rather than the example used of *namespace* = 9999999
End State	*Concept* A is *Inactive* *Concept Status* = *Moved Elsewhere* History relates *Concept* A to the Receiving Organization by use of the *Namespace Identifier*	*Concept* B is *active* *Concept Status* = *Current* History relates *Concept* B back to *Concept* A

2.3.3.7 Released Extensions

The following *extensions* are included in the *International Release* of *SNOMED CT* from the *IHTSDO*. As with any *extension*, their content may not be suitable for use everywhere, and users should consult with their *National Release* Center for information regarding the use of *extension* content within an *IHTSDO Member* country.

[4] **Assume assigned namespace = 9999999**

Table 16: Released Extensions

Extension	**Distribution**	***Extension* Contents**
U.S. Drug *Extension*	*International Release*	Actual manufactured drugs approved for distribution in the United States at the "actual medicinal product" (AMP) level. The AMP is a syntactic *normal form* consisting of: • Name (Proprietary); • Strength; • Dosage Form. All AMPs relate to "virtual medicinal product" (VMP) *concepts* in the *SNOMED CT Core*. All AMPs include the "has *active* ingredient"*relationship* where the *active* ingredient is a substance in the *SNOMED CT Core*.

2.3.4 Representational Forms for Expressions

2.3.4.1 SNOMED CT compositional grammar

The *SNOMED* Composition Grammar is a lightweight syntax for representation of *SNOMED CT expressions*. It is has been proven to be both human readable and machine parsable.

2.3.4.1.1 Background

2.3.4.1.1.1 Prior versions and status of revision

The *SNOMED* Composition Grammar was initial specified as part of the document "*SNOMED Clinical Terms Abstract Logical Models and Representational Forms, External Draft for Comment Version*". This was used extensively and was proven to be both human readable and machine parsable.

The current specification which has now been adopted as an *IHTSDO* Standard, follows the prior version in most details. It includes the following enhancements:

1. The syntax of the grammar specification is now Augmented Backus-Naur Form (ABNF) [5]which provides a formal standards-based reference for the grammar's structure.
2. Unnecessary whitespace designators, <ws>, were removed from several places in the grammar.
3. The maximum length *constraint* for *SNOMED Clinical Terms Identifiers* (*SCTIDs*) is added to this grammar. *SCTIDs* consist of sequences of digits, from a minimum of 6 to a maximum of 18 digits in length.
4. The hex code for carriage return (CR) was incorrectly given as '0C' in the previous version. It is corrected to '0D' in this version.
5. Detailed character encoding information for *UTF-8* is added.
6. The definition of *term* has been amended to allow correct parsing by the APG parser generator.

2.3.4.1.1.2 Compositional Grammar and the HL7 Code data type

The *SNOMED CT compositional grammar* allows *SNOMED CT expressions* to be represented as a text string that can be carried in *HL7* version 3 messages, in the *Code* data type. In particular, the grammar is intended to replace the *qualifier* mechanism that formerly was in the *HL7 Concept Descriptor* data type (CD data type), and which was removed in the *HL7* version 3 data types *Release* 2.

In May 2008, the *HL7* Version 3 Standard "Data Types - Abstract Specification, *Release* 2" was released for Normative Ballot 2.

[5] ABNF as defined by Internet Standard 68, RFC 5234

This revised standard defined what can be carried in the *Code* data type as "the plain code symbol defined by the code system, or an *expression* in a syntax defined by the code system which describes the *concept*."

The following details are quoted from the *HL7* Version 3 Standard: Data Types - Abstract Specification, *Release* 2, Normative Ballot 2 - May 2008 (*HL7* V3 DT R2), section 4.5.1 "Code (code): ST.SIMPLE": [6]

Table 17: Code definition from HL7 Data Types Release 2

Code (code) :*ST.SIMPLE* Definition: The plain code symbol defined by the code system, ***or an expression in a syntax defined by the code system which describes the concept.*** (emphasis added) If provided, the code SHALL be an exact match to a plain code symbol or *expression* defined by the code System. If the code system defines a code or *expression* that includes whitespace, the code SHALL include the whitespace. **An *expression* can only be used where the code System either defines an *expression* syntax, or there is a generally accepted syntax for the code System**. (emphasis added) The syntax described herein is intended to satisfy the need for a "syntax defined by the code system" as stated above, when the "code System" is *SNOMED CT*.

[6] http://www.hl7.org/v3ballot/html/infrastructure/datatypes_r2/datatypes_r2.htm

2.3.4.1.2 Compositional grammar: Normative specification

Table 18: ABNF definition of the SNOMED CT compositional grammar

expression = *concept* *("+" *concept*) [":" ws *refinements*]
concept = ws conceptId ws ["|" ws *term* ws "|" ws]
conceptId = sctId
term = 1*nonwsnonpipe *(1*SP 1*nonwsnonpipe)
refinements = (attributeSet *attributeGroup) / 1*attributeGroup
attributeGroup = "{" attributeSet "}" ws
attributeSet = attribute *("," attribute)
attribute = attributeName "=" attributeValue
attributeName = ws attributeNameId ws ["|" ws *term* ws "|" ws]
attributeValue = *concept* / (ws "("*expression* ")" ws)
attributeNameId = sctId
sctId = digitNonZero 5*17(digit)
ws = *(SP / HTAB / CR / LF) ; white space
SP = %x20
HTAB = %x09
CR = %x0D
LF = %x0A
digit = %x30-39
digitNonZero = %x31-39 ; digits 1 through 9, but excluding 0
nonwsnonpipe = %x21-7B / %x7D-7E / UTF8-2 / UTF8-3 / UTF8-4
UTF8-2 = %xC2-DF UTF8-tail
UTF8-3 = %xE0 %xA0-BF UTF8-tail / %xE1-EC 2(UTF8-tail) /
%xED %x80-9F UTF8-tail / %xEE-EF 2(UTF8-tail)
UTF8-4 = %xF0 %x90-BF 2(UTF8-tail) / %xF1-F3 3(UTF8-tail) /
%xF4 %x80-8F 2(UTF8-tail)
UTF8-tail = %x80-BF

2.3.4.1.3 Informative comments

Table 19: BNF representation of Compositional Grammar (detail)

Expression = *concept* *("**+**" *concept*) ["**:**" ws *refinements*]

<table>
<tr><td></td><td>An *expression* supports combinations of one or more *concepts* optionally refined by a set of *refinements*. The meaning of the *expression* is a *subtype* of all the *concepts* constrained by the set of *refinements*.

Note that where there is a requirement for multiple separately qualified *concepts* to be present these are expressed in *attribute groups* within a *refinement* of a general *concept* such as "*situation with explicit context*".</td></tr>
<tr><td colspan="2">*concept* = ws conceptId ws ["**|**" ws *term* ws "**|** " ws]</td></tr>
<tr><td></td><td>A *concept* is represented by a *conceptId* optionally followed by a *term* enclosed by a pair of "|" characters.

Whitespace before or after the *conceptId* is ignored as is any whitespace between the initial "|" characters and the first non-whitespace character in the *term* or between the last non-whitespace character and before second "|" character.</td></tr>
<tr><td colspan="2">conceptId = sctId</td></tr>
<tr><td></td><td>The *conceptId* must be a valid *SNOMED CT identifier* for a *concept*. The initial digit may not be zero. The smallest number of digits is six, and the maximum is 18.</td></tr>
<tr><td colspan="2">*term* = 1*nonwsnonpipe *(1*SP 1*nonwsnonpipe)</td></tr>
<tr><td></td><td>The *term* must be the *term* from a *SNOMED CT description* that is associated with the *concept* identified by the preceding *concept identifier*. For example, the *term* could be the preferred *description*, or the preferred *description* associated with a particular translation. The *term* may include valid *UTF-8* characters except for the pipe "|" character [7]. The *term* begins with the first non-whitespace character following the starting "|" character and ends with the last non-whitespace character preceding the next "|" character.</td></tr>
<tr><td colspan="2">*refinements* = (attributeSet *attributeGroup) / 1*attributeGroup</td></tr>
<tr><td></td><td>A *refinement* contains all the grouped and ungrouped attributes that refine the meaning of the containing *expression*. The ungrouped attributes, if any, are all listed first, followed by all the grouped attributes.</td></tr>
<tr><td colspan="2">attributeGroup= " **{** " attributeSet " **}** " ws</td></tr>
<tr><td></td><td>An *attribute group* contains a collection of attributes that operate together as part of the *refinement* of the containing *expression*.</td></tr>
<tr><td colspan="2">attributeSet= attribute *("**,**" attribute)</td></tr>
<tr><td></td><td>An attribute set contains one or more *attribute name* -value pairs expressing *refinements*.They are separated by commas.</td></tr>
</table>

[7] The specification for term should be comparable with the specification for the Concepts.FullySpecifiedName and Descriptions. Term fields in the release table structure (as described in SNOMED Clinical TermsTechnical ReferenceGuide, July 2008, IHTSDO). The non-pipe constraint adds greater stringency to the Compositional Grammar specification.

attribute= attributeName "**=** " attributeValue	
	An *attribute name* -value pair expressing a single *refinement* of the containing *expression*.
attributeName= ws attributeNameId ws [" **|**" ws *term* ws "**|** " ws]	
	The name (or *relationship type*) of an attribute to which a value is applied to refine the meaning of a containing *expression*. The *attribute name* is represented by an appropriate *conceptId* optionally followed by a *term* enclosed by a pair of "|" characters. Whitespace before or after the *conceptId* is ignored as is any whitespace between the initial "|" characters and the first non-whitespace character in the *term* or between the last non-whitespace character and before second "|" character.
attributeValue= *concept* / (ws "**(**" *expression* "**)** " ws)	
	A *concept* or *expression* representing the value of a named attribute which refines the meaning of a containing *expression*. If an *expression* is used this must be enclosed in brackets.
attributeNameId = sctId	
	The *attribute name* id must be the *conceptId* for a *concept* that is a *subtype descendant* of the *SNOMED CT concept*"attribute".
sctId = digitNonZero 5*17(digit)	
	A n *sctId* is used for an attribute id or a *concept* id. The initial digit may not be zero. The smallest number of digits is six, and the maximum is 18.
ws= *(SP | HTAB | CR | LF)	
	Whitespace characters (space, tab, linefeed and carriage return) are ignored everywhere in the *expression* except: **1.** Whitespace within a *conceptId* or *attributeNameId* is an error. **Note:** Whitespace before or after the last digit of a valid *Identifier* is ignored. **2.** Whitespace within a *term* is treated as a significant character of the *term*. **Note:** Whitespace before the first or after the last non-whitespace character of a *term* is ignored
nonwsnonpipe= %x21-7B / %x7D-7E / UTF8-2 / UTF8-3 / UTF8-4	
	Non whitespace includes printable ASCII characters (these are also valid UTF8 characters encoded as one octet) and also includes all UTF8 characters encoded as 2- 3- or 4-octet sequences. It excludes space (which is %x20) and the pipe character "|" (which is %x7C), and excludes CR, LF, HTAB and other ASCII control codes. SeeRFC 3629 (*UTF-8*, a *transformation* format of *ISO* 10646 authored by the Network Working Group).

digitNonZero= %x31-39	
	The first character of a *concept identifier* is constrained to a digit other than zero.
digit= %x30-39	
	Any digit 0 through 9

2.3.4.1.4 Examples of Grammar

The following examples build on each other and in complexity. They are primarily aimed at demonstrating the syntax of the *expression* grammar, although its meaning is also discussed in a number of places:

An *expression* may consist of a single *concept*, followed by a *description* associated with that *concept*. Which particular *description* to use is not mandated, but as a general rule, it may be preferable to use the *preferred term* in any particular *dialect* to achieve some level of consistency. However, such guidance is not strictly in the scope of this guide, and may be given elsewhere.

297186008 | motorcycle accident |

The syntax does not require a *description* to be associated with a particular *concept*, so the following is also a valid *expression*:

297186008

Two or more *concepts* may be combined to form a new *concept* by joining them with the "+" symbol. The resultant *expression* is the *child* of each of the *concepts* in the *expression*. The resultant *expression* below IS AN accident caused by a blizzard and also | is a | motorcycle accident.

217724009 | accident caused by blizzard | +297186008 | motorcycle accident |

Although not stipulated by the syntax, note that two *concepts* joined in this way must be from the same top level *hierarchy*. The syntax does not mandate which *concepts* in the *expression* should have associated *descriptions* and which should not so it is valid, but not advisable, to mix and match. For example, the following syntax is valid:

217724009 +297186008 | motorcycle accident |

The syntax allows spaces, tabs and carriage returns in most places. For example, the following examples have identical meaning to the one above:

217724009 + 297186008 | motorcycle accident |

217724009

+ 297186008

| motorcycle accident |

Using the "+" symbol is symmetrical and equivalent to starting with one of the *concepts* and adding an | is a | *refinement*, with a *value set* to the other *concept*. For example, the following two *expressions* are equivalent to each other and to the preceding *expression*:

217724009 | accident caused by blizzard |:

116680003 | is a | =297186008 | motorcycle accident |

297186008 | motorcycle accident |:

116680003 | is a | =217724009 | accident caused by blizzard |

One or more *refinements* may be added to a *concept* to qualify it. This is done by putting the *concept* to be qualified before a colon and the qualifying *expression* after. The qualifying *expression* is of the form "attribute = value". The example below describes an operation to remove an ovary using a laser.

83152002 | oophorectomy |:

260686004 | method |=257820006| laser excision - action |

Refinements may also be applied to a conjoined *concept*. For example, the following two *expressions* (building on a preceding example) are equivalent:

313056006 | epiphysis of ulna |:

272741003 | laterality | =7771000 | left |

119189000 | ulna part | + 312845000 | epiphysis of upper limb |:

272741003 | laterality | =7771000 | left |

Note that there are no brackets round "119189000 | ulna part | + 312845000|epiphysis of upper limb"in the above example.

Where more than one qualifying *expression* is required, these can be separated using a comma. The example below describes the removal of the right ovary using laser excision.

83152002 | oophorectomy |:

260686004 | method |=257820006| laser excision - action |,

363704007 | procedure site | =20837000 | structure of right ovary |

A further example, below, describes the removal of the left fallopian tube using diathermy excision:

120053002 | Salpingectomy |:

260686004 | method | =261519002 | diathermy excision - action |,

363704007 | procedure site | =113293009 | structure of left fallopian tube |

Where a *SNOMED CT concept comprises* a number of other *concepts* or sub - *expressions*, it may be necessary to group qualifications applied to that *concept* in order to avoid ambiguity as to how they apply. An example of a *SNOMED CT concept* that *comprises* a number of other sub - *expressions* is:

116028008 | salpingo-oophorectomy |

This procedure *comprises* two sub-procedures: the excision of part of all of the ovarian structure; and the excision of part or all of the fallopian tube structure. We should note at this point that there is a subtle difference between a *subsumptive relationship* and a *comprising relationship*:

A motorcycle accident caused by low visibility ***is a*** motorcycle accident AND

is an accident caused by a blizzard.

A salpingo-oophorectomy ***comprises*** a fallopian tube excision and an oophorectomy.

This is demonstrated by the *SNOMED CT normal form* for salpingo-oophorectomy, as shown below:

71388002 | procedure |:

{260686004 | method | =129304002 | excision - action |,

405813007 | procedure site - Direct | =15497006 | ovarian structure |}

{260686004 | method | =129304002 | excision - action |,

405813007 | procedure site - Direct | =31435000 | fallopian tube structure |}

Where it is necessary within a *postcoordinated expression* to unambiguously qualify individual components of a *concept* comprised of a number of other *expressions* (as in the above example), grouping may be used. The following example describes a salpingo-oophorectomy, with laser excision of right ovary and diathermy excision of left fallopian tube. Note that without the grouping, it would not be possible to tell on what structure the laser excision was used and on what structure the diathermy excision was used.

116028008 | salpingo-oophorectomy |:

{260686004 | method |=257820006| laser excision - action |,

363704007 | procedure site | =20837000 | structure of right ovary |}

{260686004 | method | =261519002 | diathermy excision - action |,

363704007 | procedure site | =113293009 | structure of left fallopian tube |}

A number of grouped *qualifiers* may be thus used to refine a *concept*. Note there is no comma between adjacent groups (as there are between adjacent *expressions*). Also note, the syntax does not limit the number of *qualifiers* in a group or the number of groups within an *expression*.

It is also possible to nest *expressions*, one inside the other. Any legal *expression* may be wrapped in a pair of brackets, and included in another *expression* in the same way as a *concept* would be. For example, the following *expression* describes a fracture of the femur caused by a motorcycle accident in a blizzard:

71620000 | fracture of femur |:

42752001 | due to | = (217724009 | accident caused by blizzard | +297186008 | motorcycle accident |)

In the example above, note the use of "( )" brackets, to identify a nested *expression*, as opposed to "{ }" brackets, used elsewhere, to identify groups.

The following examples show how complex *expressions* may be build up from simple ones, a layer at a time. This first *expression* describes a left hip:

24136001 | hip joint structure |:

272741003 | laterality | =7771000 | left |

This next uses the "left hip" *expression* to describe a procedure to replace it:

397956004 | prosthetic arthroplasty of the hip |:

363704007 | procedure site | = (24136001 | hip joint structure | :272741003 | laterality | =7771000 | left |)

Applying a further grouped *refinement* to the above describes a procedure to replace a left hip by inserting a prosthesis. Note that this example mixes an ungrouped qualification and a grouped qualification. Where this is done, all ungrouped qualifications should appear before the groups. Note also that there is no comma between the last qualification and the first group.

397956004 | prosthetic arthroplasty of the hip |: 363704007 | procedure site | = (24136001 | hip joint structure | :272741003 | laterality | =7771000 | left |) {363699004 | direct device | =304120007 | total hip replacement prosthesis |,

260686004 | method | =257867005 | insertion - action |}

Finally, the above *expression* may be included within a contextual wrapper, to describe a procedure that has been performed on a patient to replace a left hip by inserting a prosthesis.

243796009 | situation with explicit context |: {363589002 | associated procedure | = (397956004 | prosthetic arthroplasty of the hip |: 363704007 | procedure site | = (24136001 | hip joint structure | :272741003 | laterality | =7771000 | left |) {363699004 | direct device | =304120007 | total hip replacement prosthesis |,

260686004 | method | =257867005 | insertion - action |}), 408730004 | procedure context | =385658003 | done |, 408731000 | temporal context | =410512000 | current or specified |, 408732007 | subject relationship context |=410604004 | subject of record | }

2.3.4.2 Expression in definition forms

An *expression* can be transformed to definition form and the representations applicable to this alternative form can then be applied. However, this approach is limited because several of the forms used to represent *concept* definitions do not support nesting.

2.3.4.3 Human-readable renderings

An *expression* may be rendered according to particular rules to generate human-readable representations.

Specific "simple" rules have been specified by *NHS* Connecting for Health in the UK. Alternative suggestions for more natural rendering have also been made to extend this initial outline proposal.

Advice on this topic may be added to future revisions of this guide.

2.3.5 Stated Relationships Guide

This part of the Guide provides information about the *Stated relationship file* and the Web Ontology Language (OWL) transformation.

2.3.5.1 Stated Relationships File

The *Stated Relationship File* contains the *stated form* of *SNOMED CT*. The *stated form* of a *Concept* is the *Description Logic* definition that is directly edited by authors or editors. It consists of the stated | 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 | Is a | *relationships* and zero or more defining *relationships*.

The *Stated Relationship File* is in the same table format as the *Relationship File*, but the value of the *characteristicTypeId* field is | Stated relationship (core metadata concept) |.

The *stated form* enables implementers to test a *classifier* for consistency, by comparing the results of classification with the distributed *Relationship File*, which is the inferred form.

: Implementers should **not** use the *Stated Relationships File* unless they understand the implications of using this and provide software which makes *Description Logic* inferences from the *stated form*. The standard distribution form (the *Relationships File*) provides a *inferred view* which includes inferences derived from the *stated form*.

2.3.5.2 Description Logic (OWL or KRSS) Transform

The *Description Logic* Transform Script, written in the Perl language, performs a transform of the Stated *Relationships* into Web Ontology Language (OWL) format or KRSS format. There are two options for the syntax of the OWL output: RDF/XML, or OWL Functional Syntax. The RDF/XML is more verbose and results in a file approximately double the size of the Functional Syntax file.

2.3.5.2.1 Object Properties

SNOMED CT's attributes, the middle element of the *concept- Relationship-concept* triple, correspond to OWL Object Properties. The hierarchy under 410662002 | concept model attribute | contains all the attributes that have been approved for use as object properties. In addition, the *subtype Relationships* (i.e. | Is a | *Relationships*) between attributes in the | concept model attribute | hierarchy, as expressed in the *stated relationship file*, are used by the script to automatically generate the corresponding sub-property axioms in OWL. For example, | Procedure site - Direct | appears as a *subtype* of | PROCEDURE SITE | in the *stated relationship file*, and so the script automatically makes the OWL object property 'PROCEDURE SITE DIRECT' a sub-property of OWL object property 'PROCEDURE SITE'.

2.3.5.2.2 Relationship Grouping

When transforming *Relationships* to OWL or KRSS, all rows that have a RelationshipType that are allowed to be grouped, even if a particular row is ungrouped (i.e. even if the row has a RelationshipGroup value meaning ‘ungrouped’), must be nested under an existential restriction that represents the (potential) grouping. This existential restriction is labeled with the OWL object property called ‘Role group (attribute)’. It is just another attribute, in the sense that it has a *SNOMED CT identifier* and is named in the distributed *concept file* (609096000 | Role group |). In KRSS syntax, the stated definition of myConceptId1 with a stated definition that has a row with the triplet consisting of myConceptId1, myRelationshipType, myConceptId2 would translate into:

```
(defprimconcept myConceptId1
    (and parentConceptId
        (some RoleGroupId
            (some myRelationshipType myConceptId2))))
```

Attributes that are never grouped:

All RelationshipTypes are allowed to be grouped except | IS A |, and the following four:

- 123005000 | PART OF |
- 272741003 | LATERALITY |
- 411116001 | HAS DOSE FORM |
- 127489000 | HAS ACTIVE INGREDIENT |

2.3.5.2.3 Right Identities

There has historically been limited use of right identities, also known as property chains, in *SNOMED CT*. The one property chain that is in the current release is | DIRECT SUBSTANCE | o | HAS ACTIVE INGREDIENT | -> | DIRECT SUBSTANCE |. The OWL transform properly represents this property chain in the OWL 2 EL Profile. It is not yet represented in the *relationship file*, or anywhere else in standard *SNOMED CT distribution files*. This is a recognized deficiency which has not yet been addressed partly because there is only one such declaration, and no inferences in standard release are affected by this single right identity declaration.

2.3.5.2.4 Running the Perl transform script

Run the script according to the pattern:

```
perl <scriptfilename> <arg0> <arg1>
```

where

- <scriptfilename> is the name of the file containing the transform script
    - In the July 2014 release this Perl script file is named: tls2_StatedRelationshipsToOwlKRSS_INT_20140731.pl
- <arg0> can be KRSS, OWL, or OWLF
    - KRSS causes output to be formatted according to KRSS2 which is parsable by the OWL API 3.4.2, or by CEL or other classifiers
    - OWL causes output to be formatted according to OWL XML/RDF
    - OWLF causes output to be formatted according to the OWL functional syntax
- <arg1> is the directory containing the RF2 Snapshot subdirectories. If the current directory is RF2/Snapshot, then just use dot (".") to designate the current directory, as in the following example:

  ```
  C:\>perl tls2_StatedRelationshipsToOwlKRSS_INT_20140731.pl OWLF .
  ```

  The default output file name is snomedct_[arg0].owl, so if arg0 is OWLF, the output file will be snomedct_owlf.owl, but this can be changed in the Perl script itself (line 178) or the file can be renamed after generating it.

Alternatively you can separately supply arguments for all the file names (with their directories if necessary) :

```
perl <scriptfilename> <arg0> <arg1> <arg2> <arg3> <arg4> <arg5> <arg6>
```

- <arg0> can be KRSS, OWL, or OWLF
  - KRSS causes output to be formatted according to KRSS2 which is parsable by the OWL API 3.4.2, or by CEL or other classifiers
  - OWL causes output to be formatted according to OWL XML/RDF
  - OWLF causes output to be formatted according to the OWL functional syntax
- <arg1> is the directory path and name of the file containing the RF2 format *SNOMED CT Concept file* snapshot
  - In the July 2014 release this file is located in directory RF2Release/Snapshot/Terminology/ and is named: sct2_Concept_Snapshot_INT_20140731.txt
- <arg2> is the directory path and name of the file containing the RF2 format *SNOMED CT Description file* snapshot
  - In the July 2014 release this file is located in directory RF2Release/Snapshot/Terminology/ and is named: sct2_Description_Snapshot-en_INT_20140731.txt
- <arg3> is the directory path and name of the file containing the RF2 format *SNOMED CT Stated relationship file*
  - In the July 2014 release this file is located in directory RF2Release/Snapshot/Terminology/ and is named: sct2_StatedRelationship_Snapshot_INT_20140731.txt
- <arg4> is the directory path and name of the file containing the RF2 format Text Definitions Table snapshot
  - In the July 2014 release this file is located in directory RF2Release/Snapshot/Terminology/ and is named: sct2_TextDefinition_Snapshot-en_INT_20140731.txt
- <arg5> is the directory path and name of the file containing the RF2 Language Refset snapshot
  - In the July 2014 release this file is located in directory RF2Release/Snapshot/Refset/Language/ and is named: der2_cRefset_LanguageSnapshot-en_INT_20140731.txt
- <arg6> is the directory path and name of the output file. Any valid file name can be used.
  - for example: res_StatedOWLF_INT_20140731.owl

An example execution command on a Windows machine, from a command prompt at the directory RF2Release/Snapshot, to produce the *stated view* of *SNOMED CT* according to OWL Functional syntax, would then look like the following:

```
C:\> perl tls2_StatedRelationshipsToOwlKRSS_INT_20130731.pl OWLF
        Terminology/sct2_Concept_Snapshot_INT_20140731.txt
        Terminology/sct2_Description_Snapshot-en_INT_20140731.txt
        Terminology/sct2_StatedRelationship_Snapshot_INT_20140731.txt
        Terminology/sct2_TextDefinition_Snapshot-en_INT_20140731.txt
        Refset/Language/der2_cRefset_LanguageSnapshot-en_INT_20140731.txt
        res_StatedOWLF_INT_20140731.owl
```

2.3.5.2.5 Importing into an editor

Once the output file has been successfully created (e.g. res_StatedOWLF_INT_20140731.owl), an ontology editor that uses the OWL API should be able to import the file, assuming that the editor can handle very large files and that it is configured to use large amounts of memory, and your system has adequate memory (see FAQ below). The current version of the transform script has been tested with Protege running the OWL API version 3.4.2 and the OWL 2 Profile is OWL 2 EL. The table below presents the metrics that result from the July 2014 release.

Table 20: Metrics to Validate Import of SNOMED OWL, July 2014 International Release (20140731)

Protege Ontology Metrics	Value
Class count	299239
Object property count	62
DL expressivity	ALER
SubClassOf axioms count	229330
Equivalent class axioms count	69908
Sub object property axioms count	11
SubPropertyChainOf axioms count	1
Annotation Assertion axioms count	756457

2.3.5.2.6 SNOMED CT OWL Distribution FAQ

2.3.5.2.6.1 Access

1. Where do I obtain a copy of the OWL version of *SNOMED CT*?

- You can currently **generate** an OWL version of *SNOMED CT* using the Perl script and '*stated view*' file in the standard distribution of *SNOMED CT*.
- The Perl script and *Stated Relationships File* are distributed in the main release in different directories. The script is located in a folder called 'Resources/StatedRelationshipsToOwlKRSS/ ' and the RF2 snapshot files for *concepts*, *Descriptions* and stated *Relationships* are located in a folder called 'RF2Release/Snapshot/Terminology'. Prior to the January 2012 release, the transform was based on an RF1 format stated *Relationships* file - see documentation of prior releases for historical data and transform scripts.

2. What do you mean I need to 'generate' the OWL version of *SNOMED CT*?

- The OWL version of *SNOMED CT* is currently not distributed with the core release. However you can generate a local OWL version of *SNOMED CT* by executing the Perl script mentioned above. The instructions for using the Perl script are included in the *Stated Relationships Guide* (part of the Technical Implementation Guide), and also as comments in the header of the file containing the Perl script, which can be viewed in your favorite text editor (e.g. Notepad, Wordpad, etc).

3. What do I need to generate the OWL version of *SNOMED CT*?

- In order to generate the OWL version of *SNOMED CT*, you will need to have 'Perl' installed on your machine.
- In addition to the *Stated Relationships file* and Perl script mentioned, you will also require the RF2 (*Release Format 2*) version of the *concept file*, *description file*, and language reference set. These files are named 'sct2_Concept_Snapshot_INT_yyyymmdd.txt' , 'sct2_Description_Snapshot-en_INT_yyyymmdd.txt', and 'der2_cRefset_LanguageSnapshot-en_INT_yyyymmdd.txt' in the January 2012 *International Release* and subsequently. The first two are found in the 'RF2Release/Snapshot/Terminology' folder, and the third is found in the 'RF2Release/Snapshot/Refset/Language' folder of the *International Release*.

4. I get errors when I try to generate the OWL version using the Perl script. What can I do?

- Please check the following, before you report errors in the build process:
- Ensure you have Perl properly installed on your machine.
- Ensure that you are using versions of the Perl script, Stated *Relationships*, *Concepts*, and *Descriptions* all from the same release date and same *Release Format* (i.e. RF2). You will definitely get errors if you try to use a script designed for RF2 on RF1 format files, and vice versa. There is no guarantee of

backwards compatibility of the script - i.e. a version released for use with July 2013 RF2 files may not work with prior release RF2 files.
- Errors may be reported on the *IHTSDO Collaborative Space*, under the Implementation SIG site (in the General Discussions Forum).

2.3.5.2.6.2 Licensing

1. What are the license restrictions on the OWL version of *SNOMED CT*?

    - There is a single world-wide license for *SNOMED CT* for all purposes, called the "affiliate license". The same license applies to the OWL version of *SNOMED CT*. You can find it by following the highlighted link labelled "*SNOMED CT* Affiliate License Agreement" on the right hand side of the page at *www.ihtsdo.org/join-us/use-snomed-ct-licenses*

2.3.5.2.6.3 Importing and Visualization

1. How do I load and visualize *SNOMED CT* in OWL format?

    - Though the KRSS or OWL files generated by the Perl script can be viewed in a text editor, in order to sensibly visualize the OWL release you require a tool like Protégé 4 (http://protege.stanford.edu/). Please note that version 4 (or later) of Protégé is required to load and visualize *SNOMED CT*.

2. Protégé crashes (or becomes unresponsive) when I try to visualize the class hierarchy on my machine!

    - Protégé is known to take some time to generate the class hierarchy for display. It might be worthwhile increasing the memory allocation of Protégé before your start loading *SNOMED CT*. Please refer to the relevant Protégé documentation for exact details on increasing maximum memory available to Protégé.

3. Help, the hierarchies are rendered as *concept* IDs in Protégé! How can I change this into *fully specified names*?

    - You need change the rendering options in Protégé to render using 'labels'. In order to do that in Protégé 4, select 'Render using annotation values' from 'Preferences/Renderer/Entity rendering'.

2.3.5.2.6.4 Classification

1. What DL reasoners are currently supported for classifying OWL version of *SNOMED CT*?

    - There are Protégé 4.x plugins for several DL reasoners that can classify *SNOMED CT* provided the machine specifications listed below are met. These include Snorocket, ELK, and Fact++.

2. How long does it take to classify *SNOMED CT* in Protégé 4.x?

    - That depends on the classifier and how fast your machine is. Both Snorocket and ELK are very fast, and complete in well under 30 seconds (actual clock time) on an adequately configured machine.

3. How do I use the DL Query Tab in Protégé 4 to create *postcoordinated expressions*?

    a. We recommend looking at the Protege OWL Tutorial (http://www.co-ode.org/resources/tutorials/ProtegeOWLTutorial.pdf) for more information on using Protege 4.x to construct *expressions*. In the Protege world, *postcoordinated expressions* are referred to as DL *expressions*.

    b. In order to create *postcoordinated expressions* in the DL query tab, you are required to use the Manchester syntax for the *expressions*. In order to understand the Manchester syntax, you will need to read and work the examples in the Protege OWL tutorial.

4. What can I do once I have classified *SNOMED CT* in Protégé 4.x?

    - That depends on what you intended to do with a classified version of *SNOMED CT*. Within Protege 4.x, you can do subsumption testing over arbitrary DL *expressions* using the 'DL query tab' among other things. This feature might be used to implement subsumption testing over *postcoordinated expressions*.

2.3.5.2.6.5 Machine specification

1. What are the minimum specifications of machines for viewing loading and viewing *SNOMED CT* in OWL?
    - As a general rule, for reasonable performance, one would require a 64-bit machine, such as an Intel Core 2 Duo, with clock speed of 2GHz or more and 4GB of RAM to load the OWL version of *SNOMED CT* in Protégé.
    - The actual memory requirements might actually be smaller depending on your machine. Users have successfully loaded *SNOMED CT* on a 32-bit Mac OS X machine with 2GB RAM, and on a 32-bit Linux (Ubuntu) machine with 3GB RAM. However, display and editing performance is usually considered unacceptably slow when using these minimal configurations.
    - Loading and visualizing the OWL version of *SNOMED CT* using alternate methods might have different machine specifications.
2. What are the minimum specifications for classifying *SNOMED CT*?
    - It is believed that one would require a 64-bit machine with an Intel Core 2 Duo processor (or better) with 4GB of RAM to classify *SNOMED CT* using the classifiers bundled with Protégé 4. Users have successfully classified *SNOMED CT* on a 32-bit Mac OS X machine with 2GB RAM, and on a 32-bit Linux (Ubuntu) machine with 3GB of RAM.

2.3.5.2.6.6 Software

1. Can I bundle the OWL version of *SNOMED CT* in my open source software?
    - *SNOMED CT* is licensed under the affiliate license described above. *SNOMED CT* or any derivatives of *SNOMED CT* cannot be redistributed under any other license (including any form of open source license).
2. Am I allowed to make extensions or modification to the OWL release of *SNOMED CT* and include it in my software?
    - *SNOMED CT* is licensed under the affiliate license described above. *SNOMED CT* or any derivatives of *SNOMED CT* cannot be redistributed under any other license (including any form of open source license).
3. What API can I use to programmatically access the OWL version of *SNOMED CT*?
    - Though there are many candidate *APIs* available, most DL reasoners bundled with Protégé 4.x use the Manchester OWL API (*owlapi.sourceforge.net*). There are examples online on how to load an ontology. It might also be possible to use the Jena API (*jena.sourceforge.net*) to load the RDF/XML version of the file.

2.3.6 Other Representational Forms

This section summarizes some of the other forms in which *SNOMED CT components* and *expressions* may be represented. This includes some references to a selection of proprietary and standard representation which have been used or suggested for particular uses. Mention in this section is intended to be illustrative and does not represent endorsement. Additional suggestions that may be helpful to some implementers could be added in future.

2.3.6.1 Complete Concept Representations

Representation of the *concept* as a whole includes the definition of the *concept* but also includes additional properties of *concepts* and associated components such as *descriptions* and *Reference Sets*.

As a rule representations of complete *SNOMED CT concepts* will be specific to *SNOMED CT*. Some of these representations will be specified by *SNOMED* and others will be application specific designs building on the *SNOMED CT* specifications. If generic forms of representation are used then guidelines on how particular properties from *SNOMED* are represented are necessary.

2.3.6.1.1 SNOMED CT distribution files

The *Release File Specifications* provide a form of representation for complete *concepts* (and other *components*).

The *release files* are designed efficient for large scale batch distribution and facilitate easy import into relational databases. They may need to be indexed and optimized to provide a practical implementable representation.

2.3.6.1.2 IHTSDO workbench internal format

The set of database table used by the *IHTSDO Workbench* to maintain *SNOMED CT* include a full representation of all types of *SNOMED CT Components*. The representation is closely aligned with *SNOMED CT Release Format 2*. However, additional data is stored to manage workflow and conflict resolution during the development process.

2.3.6.1.3 SNOMED CT Distribution XML

The XML distribution schema specified by *SNOMED* provides a form of representation for complete *concepts*(including associated components).

The XML distribution files can be used as an alternative to the *SNOMED CT distribution files*. However, they are particularly efficient for communication of individual *concepts* or sets of *concept* (e.g. for update change-sets).

2.3.6.1.4 Application internal

SNOMED CT enabled applications will usually have their own internal optimized representation of the *SNOMED* distribution information. This may simply be a relational database with a specified set of indices or it may be a significantly different form.

Examples of proprietary representation include the forms used internally by CliniClue (ClueData), Health *Language*, Apelon TDE and other implementations.

2.3.6.1.5 Various human-readable renderings

Concept information may be rendered in various ways to allow human visualization and understanding. These forms may include plain text, mark-up and graphical trees diagramming *relationships*. All of these renderings can be regarded as representations of complete *concepts* or their definitions.

2.3.6.2 Concept Definition Representations

See also: *Complete concept representations*

2.3.6.2.1 KRSS

KRSS is a general form for representing logical *descriptions*.

Transforms have been developed internally for producing KRSS representations of *SNOMED CT* definitions (see *Stated Relationships Guide*).

2.3.6.2.2 OWL

The Web Ontology Language (OWL) is a web-technology based approach to representation of logical *concept definitions*.

Transforms have been developed internally for producing OWL representations of *SNOMED CT* definitions (see *Stated Relationships Guide*).

2.3.6.2.3 Representing Definitions as Expressions

A *Concept* definition can also be represented as an *expression* (see *Representational Forms for Expressions*). One or more of the *supertype parent concepts* are represented as *focus concepts* and other defining *relationships* are represented as refining *attributes*.

2.3.6.2.4 Various human-readable renderings

*Concept definitions*may be rendered in various ways to allow human visualization and understanding. These forms may include plain text, mark-up and graphical trees diagramming *relationships*. All of these renderings can be regarded as representations of *concept definitions*.

2.3.7 Additional Reference Materials

2.3.7.1 Introduction

This section contains additional technical information that is referenced by other parts of this guide.

2.3.7.2 Unicode UTF-8 encoding

2.3.7.2.1 Introduction

UTF-8 is an efficient encoding of *Unicode* character - *strings* 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 issupported by all widely used operating systems and major applications (and has been for more than 15 years).

SNOMED CT uses the *UTF-8* representation of characters in *terms* and other text fields.

2.3.7.2.2 Character encoding

ASCII characters are encoded as a single byte.

- Greek, Hebrew, Arabic and most accented European characters are encoded as two bytes;
- All other characters are encoded as three bytes;
- The individual characters are encoded according to the following rules.

2.3.7.2.2.1 Single byte encoding

Characters in the *range* 'u+0000' to 'u+007f' are encoded as a single byte.

Table 21: UTF-8 Single Byte Encoding

byte 0	
0	bits 0-6

2.3.7.2.2.2 Two byte encoding

Characters in the *range* 'u+0080' to 'u+07ff' are encoded as two bytes.

Table 22: Two byte encoding

byte 0				byte 1		
1	1	0	bits 6-10	1	0	bits 0-5

2.3.7.2.2.3 Three byte encoding

Characters in the *range* 'u+0800' to 'u+ffff' are encoded as three bytes:

Table 23: 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

2.3.7.2.3 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 24: 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 in *UTF-8*

2.3.7.2.4 Example encoding

Table 25: UTF-8 Encoding Example

Character	S	C	T	®				
Unicode	0053	0043	0054	00AE		2462		
Bytes	01010011	01000011	01010100	11000010	10101110	11101111	10111111	10111111

2.3.7.3 Check-digit Computation

The *SCTID* (See Component features - Identifiers on page 20) 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.

2.3.7.3.1 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 26: 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
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 27: 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

	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
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 28: 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*).

  .

2.3.7.3.2 Sample Java Script for computing Verhoeff's Dihedral Check

The script is presented here as part of an HTML page.

Note:

The code below can be used by copying all the lines in the above section into an HTML file and opening this with a web *browser*. From the HTML version of this guide the following link provides access to this file as an *web page*.

```
<!DOCTYPE html SYSTEM "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html>
   <head>
      <title>SNOMED CT Identifier Check</title>
      <style>
      body{font-family:Arial, Helvetica, sans-serif}
}
      </style>
      <meta content="text/html; charset=iso-8859-1" http-equiv="Content-Type"><meta>
         <script type="text/javascript" language="JavaScript">
```

```
        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";
    }
            </script>
    </head>
    <body>
        <h1>SNOMED CT Identifier Check</h1>
        <form action="" name="form">
            <table border="1" width="441">
            <tr>
                <td width="212" height="25">
                    Partial Identifier <br/>(without check-digit) 
                </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 
            </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>
    </body>
</html>
```

2.3.7.3.3 Sample Visual Basic for computing Verhoeff's Dihedral Check

```
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 & 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
```

2.3.7.3.4 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.

2.3.7.3.4.1 A brief comparison of check-digit effectiveness

2.3.7.3.4.1.1 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.

2.3.7.3.4.1.2 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 each *namespace*. More significantly the unusable item *Identifiers* 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.

2.3.7.3.4.1.3 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* methods 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).

2.3.7.4 Search Support Tables

2.3.7.4.1 Overview

Effective implementation of *SNOMED CT* depends on the ease and speed with which users can locate the *terms* and *Concepts* that they wish to use. An essential contribution to meeting this requirement is the ability to perform rapid and flexible text searches.

A set of word search tables (indexes) are included in the Developer Toolkit. These tables are designed to facilitate development of effective search facilities while reducing duplication of effort. However, neither these tables, nor indices derived from them, are sufficient to meet the full range of search requirements. Meeting the needs of different users for appropriate methods for locating particular *Concepts* is an area in which competitive development is expected and welcomed. Developers may choose to use some or all of the word search tables distributed with *SNOMED CT* or may develop their own solutions independent of these tables.

The intention of the word search tables is to identify candidate matches among the *Descriptions* (or *Concepts*) of *SNOMED CT*. An application or coding engine will apply further filtering to these candidate matches to identify the matches to be selected or displayed. A balance must be made between specificity and completeness of a search. The *keyword* algorithm is intended to maximize the likelihood that the required *Concept* will be included in the candidate matches rather than to achieve precision.

Applications may filter candidate matches using techniques that are many and varied. Some may take account of non-textual characteristics (e.g. *Reference Sets*, *subtype Relationships* or *Relationships*) while others use more complex textual techniques (e.g. word *order* dependence, case dependence, complete phrase matching, regular-*expression* matching, Soundex). These extended text search techniques are beyond the scope of the *keyword* generation algorithm.

The algorithm for *keyword* generation is only applicable for English and other western European *languages*. It is not intended to apply to Russian, Greek, Slavic or to any non-European *languages*.

Please refer to the *Technical Implementation Guide* for additional search implementation guidance.

2.3.7.4.2 Search index - structure diagram

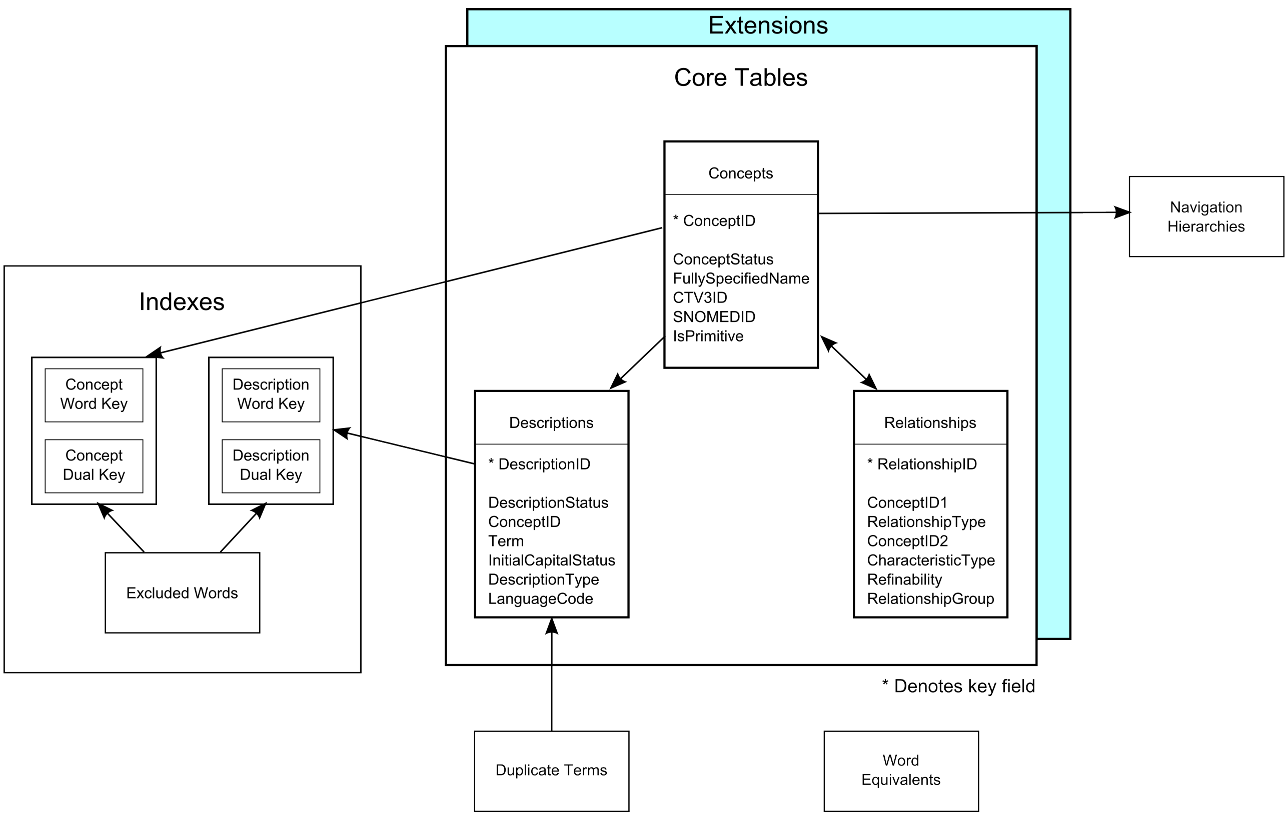


Figure 32: Search Index Overview (RF1)

2.3.7.4.3 Word Search Tables - Summary

The following five tables are included in the *Developer Toolkit* of *SNOMED CT*. These tables are derived from the *SNOMED CT Descriptions* Table. The *LanguageCode* of the *Descriptions Table* is used to choose only *descriptions* for a *language*.

Table 29: Summary of Word Search Tables

Table	*Description*
Excluded Words Table	Each row in this table is a word excluded from the list of possible *keywords* and *dualkeys*. Words are excluded if they are frequently used and are so limited in semantic specificity that they impair rather than enhance searches.
DescWordKey Table	Each row in this table is a word followed by a reference to a *Description* in which this word appears.
ConcWordKey Table	Each row in this table is a word followed by a reference to a *Concept*. A *Concept* is referenced if the word appears anywhere in the combination of the *Fully Specified Name* with the *current* valid *Preferred Term* and *Synonyms*.
DescDualKey Table	Each row in this table is a six-character *string* representing the first three letters of a pair of words followed by a reference to a *Description* in which these two words appear.
ConcDualKey Table	Each row in this table is a six-character *string* representing the first three letters of a pair of words followed by a reference to a *Concept*. A *Concept* is referenced if both words appear anywhere in the combination of the *Fully Specified Name* with the *current* valid *Preferred Term* and *Synonyms*.

All *keywords* are regarded as case independent and are presented in the word search tables in upper case. Case dependent searching can be applied by appropriately filtering the candidate matches.

2.3.7.4.4 Word Equivalents

The *Word Equivalent* Table is included in the Developer Toolkit of *SNOMED CT*. It supports enhanced searches that take into account semantically similar words such as KIDNEY and RENAL. It also provides commonly used abbreviations. This table can be used by implementers to offer additional search capability in applications without greatly increasing the volume of *synonyms*. It is not intended as a comprehensive dictionary of words. Many searches can be completed without using this table; like the other word search tables, it is completely optional and can be used as an example of a capability that may be customized and extended by *SNOMED CT* implementers.

2.3.7.4.4.1 Word Equivalents Tables - Summary

Table 30: Word Equivalents Table

Key Fields	
WordBlockNumber	A 32-bit *integer* shared by a set of equivalent words or phrases. The *WordBlockNumber* links together several rows that have an identical or similar meaning.

Key Fields	
WordText	A word, phrase, acronym or abbreviation that is equivalent to the *WordText* of other rows that share the same WordBlockId.

Data Fields	
WordType	An *integer* indicating the type of *equivalence*
WordRole	An *integer* indicating the usual role of this word. This should be considered if attempting to find a *postcoordinated* combination of *Concepts* that matches a phrase.

Chapter 3

3 Release File Formats

Currently, during a transitional period, there are two distinct *Release Formats* for *SNOMED CT*:

- *Release Format 2* (*RF2*): The new standard distribution format for *SNOMED CT*. This was developed in response to extensive feedback on its predecessor and will replace *RF1* during 2012.
- *Release Format 1* (*RF1*): The specification in which *SNOMED CT* has been provided since its first release in 2002. This format will be phased out, but support of applications that require *RF1* format files will be available using a conversion application developed and supplied by the *IHTSDO*.

The key enhancements in *RF2* are:

- More robust and consistent version representation;
- *Reference sets*, provides a more easily extensible and maintainable replacement for *RF1 subsets* and crossmaps;
- Use of an added *hierarchy* to represent metadata about the structure of *SNOMED CT* itself.

Both *Release Formats* represent:

- The components of *SNOMED CT*:
  - *Concepts*
  - *Descriptions*
  - *Relationships*
- Additional *derivatives* that provide standard representations of :
  - Value-sets consisting of a specified set of *concepts* or *relationships*
  - *Cross Mapping* tables to other codes and classifications.

Both *Release Formats* are provided in:

- Tab-delimited text files;
- Represent character content in accordance with the *Unicode UTF-8* specification;
- Use *SNOMED CT Identifiers* as the permanent *Identifier* of released core components;
- Support *extensions* to the *International Release* using *namespaces* allocated to licensees to denote the provenance of added components and to ensure *Identifier* uniqueness.

3.1 RF1 Compatibility and Conversion Tools

In January 2012 the *IHTSDO* 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 *IHTSDO* 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 *IHTSDO* 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 *Identifiers* of RF1 derivatives (Subsets and *Cross Maps*) differ from those used for the equivalent *Reference Sets* in RF2. These differences are an essential consequence of ensuring that the RF1 data produced by conversion from RF2 is fully compatibility with existing RF1 systems.
- The RF2 to RF1 Conversion Tool used for generating the RF1 files is also available to all *IHTSDO 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 *IHTSDO 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.

3.2 Release Format 1

This section describes the file structures in which *SNOMED Clinical Terms* has been distributed since its initial release in 2002. These file structures, which are now referred to as *Release Format 1* (*RF1*), are being replaced by *Release Format 2* which offers a range of enhanced features to support effective implementation and maintenance. For more details of see the *Release Format 2 specification*.

From 2012 *RF2* will become the primary *Release Format* for *SNOMED CT*. The *IHTSDO* will provide an *RF2* to *RF1* conversion application for use by Members and Affiliates to support existing applications that depend on the *RF1* during a transitional period. However, this conversions process cannot support many of the enhanced features of *RF2* and for this reason implementer are recommended to migrate their applications to the newer format.

3.2.1 Release Format 1 - Overview

The tables and related mechanisms used to represent *SNOMED CT Components* and *Derivatives* in *Release Format 1* are summarized below.

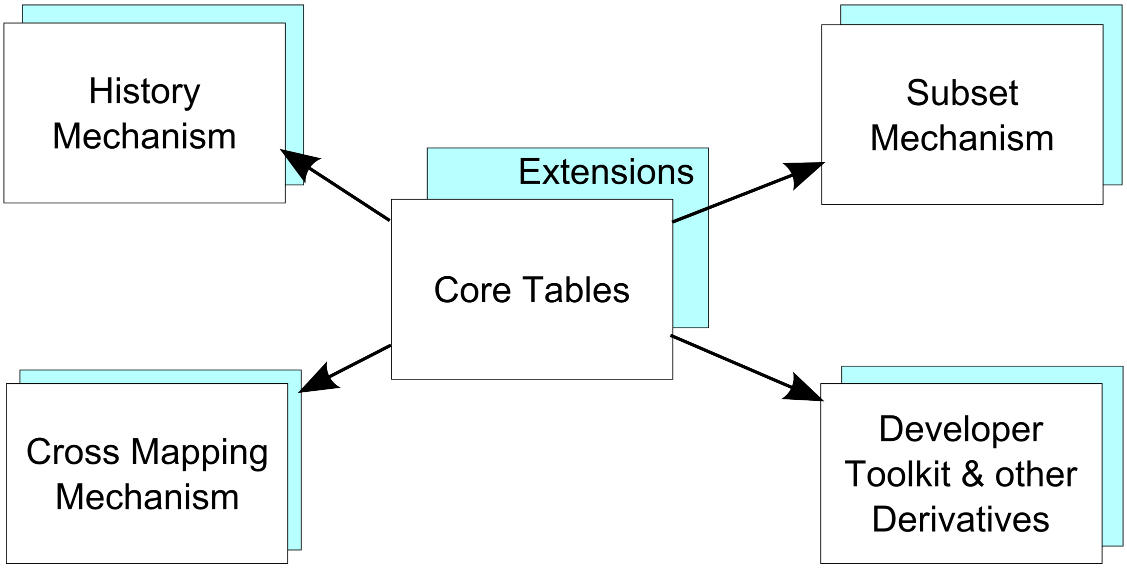


Figure 33: SNOMED CT Data Structure Overview

- ***Core Tables***: The *Core Tables* contain the *Concepts*, *Descriptions*, and *Relationships* of the *SNOMED CT* terminology. The other structural mechanisms support and enrich the *core table* structure for terminology implementers. Briefly, these other mechanisms include.
- **History**: History files are typically useful for upgrading an implementation of *SNOMED CT* to a new *SNOMED CT release*. History files include a log of *Concept* and *Description* additions, inactivations and minor changes. History files also include information about what *concepts* can be used in place of inactivated (*retired*) *concepts*.
- ***Subsets***: *Subsets* define a smaller collection of *SNOMED CT Concepts*, *Descriptions*, or *Relationships*.
- ***Cross Mappings***: *Cross Mappings* relate *SNOMED CT* to a target classification or coding scheme such as ICD-9 or *ICD-10*, in order to allow *SNOMED CT* -encoded data to be expressed using the *target scheme* for a particular purpose or use case.
- ***Extensions***: *Extensions* consist of terminology content developed to supplement, but not replace, the content contained in the *International Release*. *Extension* content may be used to meet requirements that are not sufficiently universal to justify the creation of *International Release* content. An example would be a national drug *extension*, containing *concepts* for the proprietary drugs available within a particular country. *Extensions* may be developed and maintained by the *IHTSDO*, a *National Release* Center or an authorized
.
- **Developer Toolkit** : Tables in the Developer Toolkit can be helpful for software developers directly or as examples that can be customized by installation and include:
  - Indexes, *Word Equivalents*, and *Duplicate Term* files that are helpful for search applications;
  - Sample *Navigational Hierarchies* that can be used to manage the display of *SNOMED CT concepts* and viewing sequences.
- ***Canonical Table***: The *canonical table* is useful for determining the logical *equivalence* of *concepts* that may be represented in multiple ways.

This guide provides details about the *SNOMED CT* data structures. This section provides an overview of the structure and answers some of the frequently-asked questions which may be helpful if you are new to *SNOMED CT*.

3.2.2 Release Format 1 - Structure diagram

The diagrams in this section summarize the data structures using in *SNOMED CT Release Format 1*. The next section provides the formal specifications of the distributed files.

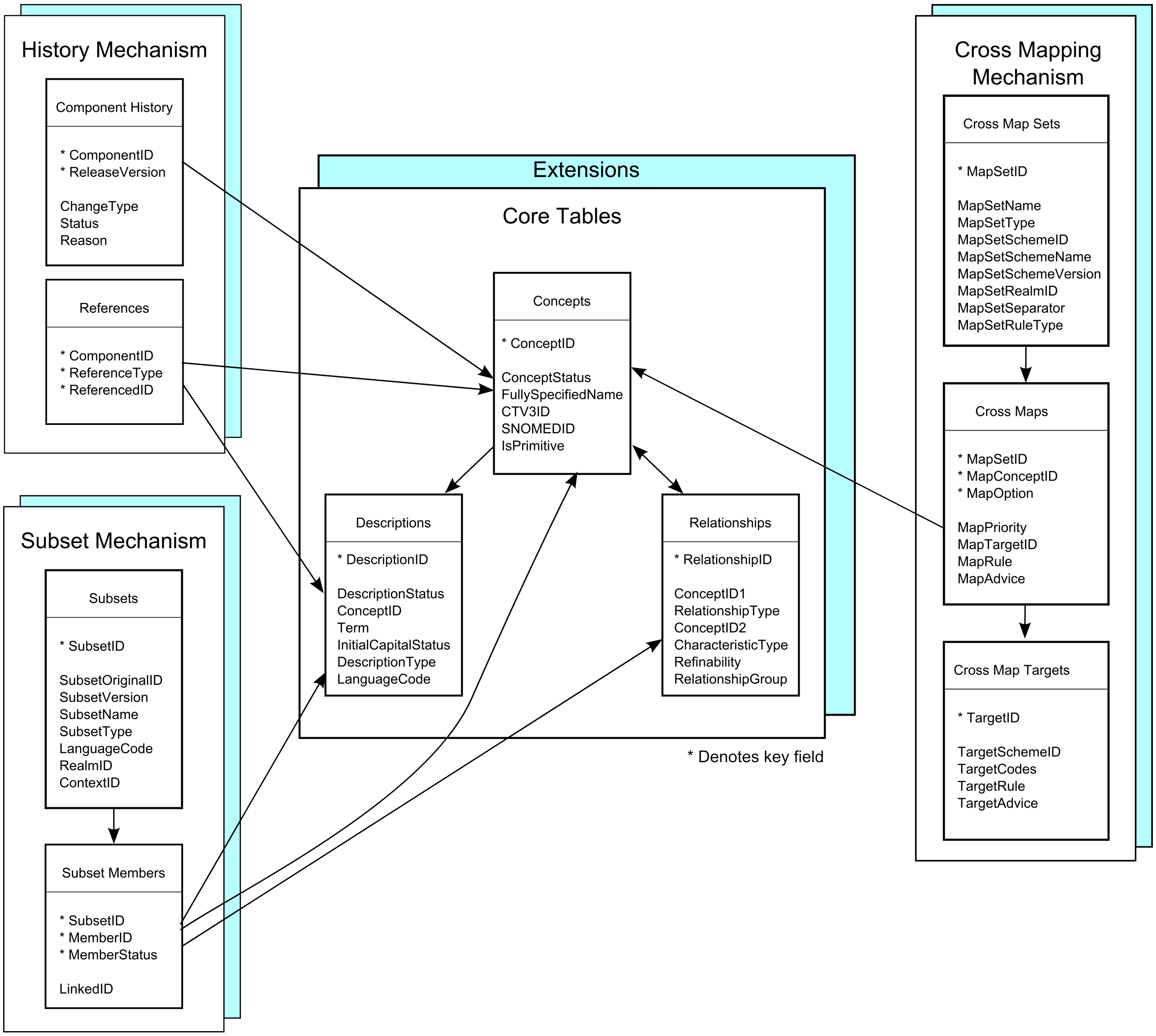


Figure 34: Release File Overview (RF1)

3.2.3 Core Table Structure

This section focuses upon the three core components of *SNOMED Clinical Terms*:

1. Clinical *concepts*
2. *Terms* used to describe *concepts*, and;
3. *Relationships* between *concepts*.

This section describes the logical structure of these elements and the physical structure of the files in which they are distributed. See *Release Format 1 - Detailed specification* on page 138 for full details.

3.2.3.1 Overview

The core structure of *SNOMED Clinical Terms* ® comprises the following three tables:

Table 31: Core Table Summary

Concepts Table	Each row in this table represents a clinical *concept*.
Descriptions Table	Each row in this table specifies a *term* that can be applied to describe a single clinical *concept*.
Relationships Table	Each row in this table specifies a *relationship* between two clinical *concepts*. The nature of each *relationship* is represented using a special kind of clinical *concept*. The figure below illustrates the *relationships* between these tables.

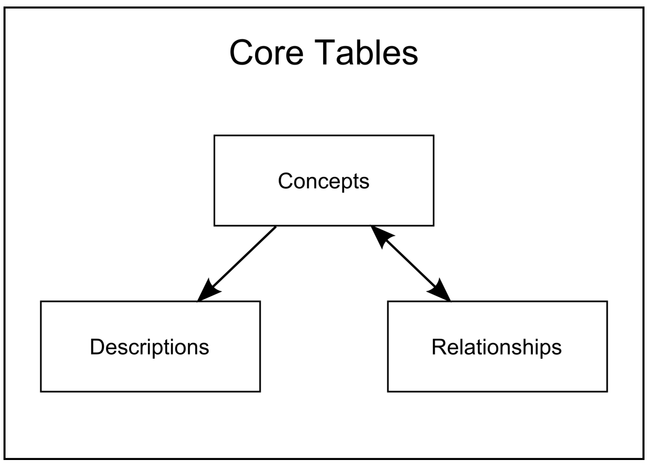


Figure 35: SNOMED CT core Structure Overview

- A *concept* is described by the *term* in 2-n *descriptions*
- Each *description* refers to one *concept*
- A *relationship* refers to three *concepts*: a source, target, and *relationship type*
- A *concept* is the source of 1-n *relationships* (except the *root concept*);
- A *concept* is the target of 0-n *relationships*
- A *concept* represents the type of *relationship*

3.2.3.2 Core Tables Data - Structure diagram

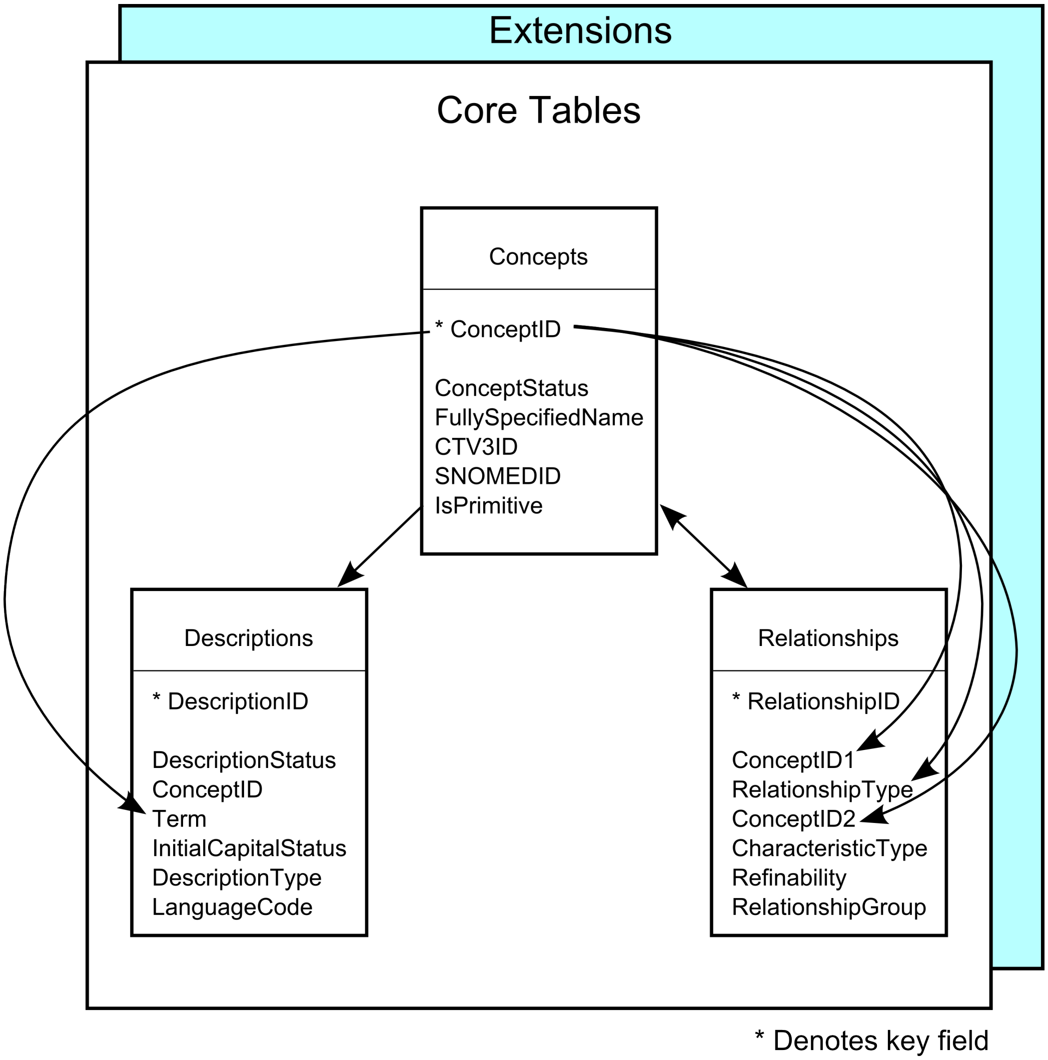


Figure 36: Core Table Summary (RF1)

- A *Concept* is described by the *Term* of 2-n *Descriptions*;
- A *Relationship* refers to 3 *Concepts*: a source, target, and *relationship type*;
- Each *Description* refers to 1 *Concept*;
- A *Concept* is the source of 1-n *Relationships* (except the root *Concept*);
- A *Concept* is the target of 0-n *Relationships*;
- A *Concept* represents the type of *Relationship*.

3.2.3.3 Core Tables - Summary

3.2.3.3.1 Concepts Table - Summary

Each row in the *Concepts Table* represents a clinical *concept*. Each *Concept* has a unique *Identifier* and a *Fully Specified Name* specifying the nature of the *concept*.

Table 32: Concepts Table

Key Fields	
ConceptId	The unique *SNOMED CT Identifier* for this *Concept*.

Data Fields	
ConceptStatus	The *status* of a *Concept* indicates whether it is in *active* use and, if not, indicates the *reason* for withdrawal from *current* use.
FullySpecifiedName	A unique phrase that describes a *Concept* in a way that is intended to be unambiguous. The *Fully Specified Name* is also present in the *Descriptions Table*. It is not the same as the *Preferred Term*, which is also in the *Descriptions Table*. The *Fully Specified Name* explains the meaning of the *concept* more fully than the *Preferred Term* to remove or reduce ambiguity.
CTV3ID	The *Read Code* for this *Concept*. **Note:** The *CTV3ID* field should no longer be relied upon for mapping to and from the *Read Codes*. Additional mapping work in the UK identified some anomalies and resulted development of more flexibility table for *Read Code* Mapping
SNOMEDID	The legacy SNOMED International 3 to 3.5 code for the *Concept*.
IsPrimitive	Indicates whether a *Concept* is *Primitive* or *Fully defined* by its *current* set of *Defining characteristics*.

3.2.3.3.2 Descriptions Table - Summary

Each row in the *Descriptions Table* associates a *term* with a clinical *concept*, which it can be used to represent. Each *Description* has its own unique *Identifier* and also contains the text of a *Term* and the *Identifier* of the *Concept* it may represent.

Table 33: Descriptions Table

Key Fields	
DescriptionId	The unique *SNOMED CT Identifier* for this *Description*.

Data Fields	
DescriptionStatus	The *status* of a *Description* indicates whether it is in *active* use and, if not, indicates the *reason* for withdrawal from *current* use.
ConceptId	The unique *SNOMED CT Identifier* of the associated *Concept*.
Term	The text of a *Term* used to describe the associated *Concept*.

Data Fields	
InitialCapitalStatus	An indication of whether the capitalization *status* of the first character of the *Term* is significant.
DescriptionType	An indication of whether the *Term* is the *Fully Specified Name*, *Preferred Term* or *Synonym* for the *Concept* to which this *Description* applies.
LanguageCode	An indication of a *Language* or *Dialect* in which this *Description* is valid. The *language* or *dialect subset* ultimately defines the *descriptions* for each *concept*.

To identify the *descriptions* for any *language* edition, the appropriate *Language Subset* must be used. Do not use the *Descriptions Table* alone. The *descriptions* for each *language* edition (including English) are designated in the *Subset*.

3.2.3.3.3 Relationships tables - Summary

Each row in the *Relationships Table* represents a *Relationship* between two *Concepts*. The type of *relationship* is identified by reference to another *Concept*.

Table 34: relationships tables

Key Fields	
RelationshipId	The unique *SNOMED CT Identifier* of this *Relationship*.

Data Fields	
ConceptId1	The unique *SNOMED CT Identifier* of the *Concept* which is the source of this *Relationship*.
RelationshipType	The unique *SNOMED CT Identifier* of the *Concept* which represents the type of *relationship* between the related *Concepts*.
ConceptId2	The unique *SNOMED CT Identifier* of the *Concept* which is the target of this *Relationship*.
CharacteristicType	An indication of whether a *Relationship* specifies a *defining characteristic* of the source *Concept* or a possible *qualifying characteristic* of that *Concept*.
Refinability	An indication of whether it is possible to refine the target *Concept* when this *relationship* is used as a template for clinical data entry.
RelationshipGroup	An *integer* value that links together *Relationships* which are part of a logically associated *Relationship group*.

The *Relationships Table* is concerned with definitions, *qualifiers* and additional facts about a *Concept*. Although it can also be used to display a *hierarchy* of *subtypes* it does not have a specified or natural display *order*. Ordering is supported by use of a *Navigation Subset* (see *Subset Mechanism* on page 109.

In 2003-2004, some *relationship types* were packaged into a separate file called the *Historical Relationships* Table. These *relationship types* were merged into the *Relationships Table* starting with the January 2005 release.

3.2.4 Subset Mechanism

A *Subset* refers to a set of *Concepts*, *Descriptions*, or *Relationships* that are appropriate to a particular *language*, *dialect*, country, specialty, organization , user or context.

In its simplest form, the *Subset* Mechanism is a list of *SNOMED CT Identifiers* (*SCTIDs*). Each *SCTID* refers to one component of *SNOMED CT* that is a member of the *Subset* (called a "*Subset Member*"). As an analogy, think of *SNOMED CT* is a book. A *Subset* is like an index entry pointing to a set of pages relevant to a particular topic.

The *Subset* Mechanism may be used to derive tables that contain only part of *SNOMED CT*. In some cases, these derived tables may also be centrally distributed (e.g. a release table containing only *Descriptions* for a particular *International Edition*). Please refer to *Released Subsets* on page 113 for a list of *subsets* now available.

SNOMED CT provides for the *Subset Types* listed below. Some of these types are not yet in use.

3.2.4.1 Subset Mechanism - Structure diagram

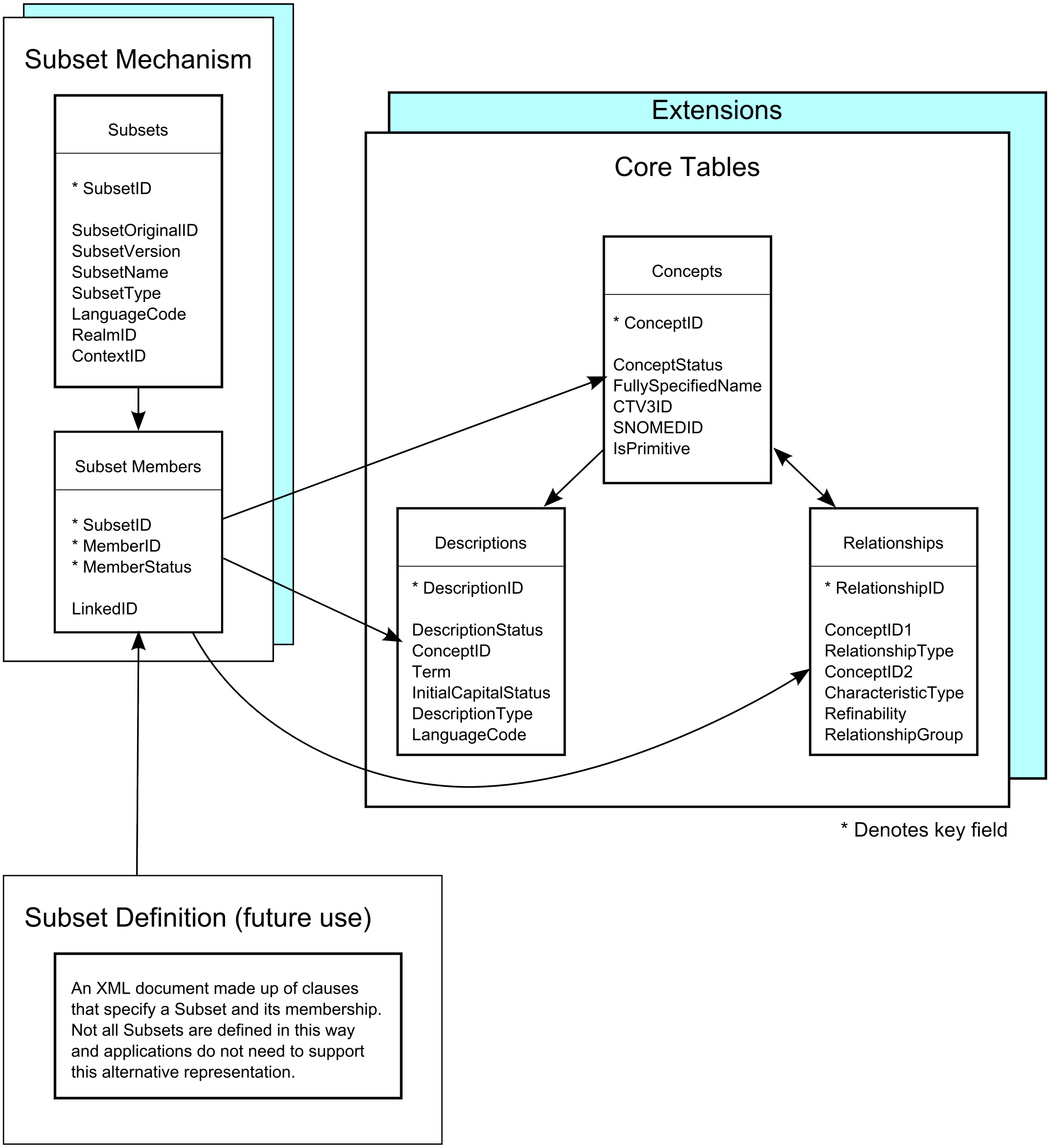


Figure 37: Subset Mechanism Overview (RF1)

A *Concept* may be a member of 0-n *Subsets*.

A *Description* may be a member of 0-n *Subsets*.

A *Relationship* may be a member of 0-n *Subsets*.

A *Subset Member* may refer to 1 *Concept*, 1 *Description*, or 1 *Relationship*.

MemberID may be a *ConceptId*, *DescriptionId*, or *RelationshipId*, depending on the *SubsetType*.

A *Subset Definition* may specify the membership of a *Subset*.

3.2.4.2 Subset Tables - Summary

Subsets provide a way for users and implementers to limit the set of *SNOMED CT Concepts* and *Descriptions* for a particular purpose. There are several types of *subsets* already defined for specific goals, and other types will be utilized in the future.

A common file structure is used for all *Subsets*. This approach simplifies the release structure and installation process for all *SNOMED CT* users.

Subsets are released using two tables:

Table 35: Subset Mechanism Summary

Subsets Table:	Each row in this table describes one release of a *Subset*. This table includes *SNOMED CT Subsets* that are packaged together in the *Subset Members* table.
Subset Members Table:	Each row in this table represents one member of a *Subset*. The member may be a *Concept* or a *Description*. One or more *Subsets* may be packaged together in this table.

3.2.4.2.1 Subsets Table - Summary

Each row in the *Subsets Table* describes a *Subset* and characteristics of that *Subset*.

Table 36: Subsets Table

Key Fields	
SubsetId	The unique *SNOMED CT Identifier* for this *Subset*.

Data Fields	
SubsetOriginalId	The unique *SNOMED CT Identifier* for the original *Subset* of which this *Subset* is a version.
SubsetVersion	An *integer* incremented for each revised release of a *Subset*.
SubsetName	A name that describes the purpose or usage of this *Subset*.
SubsetType	Indicates the nature of the *Subset* and the type of *SNOMED CT Component* that may be a member of the *Subset*.
LanguageCode	Identifies the *Language* and optionally the *Dialect* to which the *Subset* applies (only used for *description* -based *subsets*: *Language*, *Realm Description*, and *Realm Concept*).
RealmId	Identifies the *Realm* to which the *Subset* applies.

Data Fields	
ContextId	May identify the *Context Domain* to which the *Subset* applies.

3.2.4.2.2 Subset Members Table - Summary

Each row in the *Subset Members Table* sets the *status* of a member of an identified *Subset*.

Table 37: Subset Members Table

Key Fields	
SubsetId	The unique *SNOMED CT Identifier* for this *Subset*.
MemberId	The *SNOMED CT Identifier* of this *Subset Member*. This may be a *ConceptId*, *DescriptionId* or *RelationshipId*.

Data Fields	
MemberStatus	An *integer* specifying the *status*, type or *order* of this member.
LinkedId	Valid for *Navigation* and *Duplicate Terms Subsets* only. For *Navigation Subsets* it is the *SNOMED CT Identifier* for a *Concept* that is a *Navigation child* of the *Subset Member*. For *Duplicate Terms Subsets* it is the *SNOMED CT Identifier* for the highest priority *Description* sharing the *Duplicate Term*.

3.2.4.2.3 Subset Definition File - Summary

The *Subset Definition File* is an XML document that contains the definition of a *subset*. The definition consists of a set of "rules" that can be applied to *SNOMED CT* to determine the membership of the *subset*, rather than having each member identified separately. The rules are expressed as clauses that contain tests for each *concept* or *description*. There are three types of tests:

- Hierarchical Selection

  This test identifies *concepts* that are *subtypes* (*descendants*) or supertypes (*ancestors*) of a specified *concept*. For example, create a *subset* that contains "infectious disease' and all its *subtypes*.

  .
- *Relationship* Selection

  This test identifies *concepts* that have a specified attribute and value. For example, create a *subset* that contains all *concepts* where the Finding Site (attribute) = "heart structure' (value).

  .
- Property Selection

  This test identifies *concepts* that match a property value in the *SNOMED CT release* tables. For example, create a *subset* that contains all *concepts* with a *Status* = 0 (*current concepts*) or *concepts* that contain the text *string* "K deficiency.'

  .

The *Subset Definition File* allows multiple tests to be used to define a *subset*, and to use true/false conditions to be specified to determine which tests, and in which sequence, the tests should be used. The tests determine the membership of the *subset* and the value of the Member *Status* field.

The *Subset Definition File* also contains metadata about the *subset*, such as the *Subset Identifier*, the *Subset Version*, the *LanguageCode*, and *Realm Identifier*.

The *Subset Definition File* can be used to represent *concept* and *description subsets*. It cannot be used for *navigation subsets*, or for *duplicate terms subsets*.

Future specifications for the *Subset Definition* Filemay enable multiple *subsets* to be specified in one file, and for one *subset* to be used as the basis for another *subset* so that only the differences between the *subsets* need to be specified.

3.2.4.3 Released Subsets

The following *subsets* are maintained and licensed by the *IHTSDO*. The date of first issue and, where appropriate, retirement are noted.

Table 38: Released Subsets

Subset	**Type of *Subset***	***Subset* Contents**
US English *Dialect Subset*	*Language* / *Dialect*	Identifies *Descriptions* that are appropriate for the US *dialect* of the English *Language*. This *subset* applies to the English *Language Descriptions Table*. This *Subset* was first available with the First *Release* US Edition.
UK English *Dialect Subset*	*Language* / *Dialect*	Identifies *Descriptions* that are appropriate for the British *dialect* of the English *Language*. This *subset* applies to the English *Language Descriptions Table*. This *Subset* was first available with the First *Release* UK Edition.
Spanish *Subset*	*Language*	Identifies *Descriptions* that comprise the Spanish Edition of *SNOMED CT*. This *Subset* is required to determine the contents of the Spanish release, which includes *descriptions* both in English and Spanish. This *subset* applies to the Spanish Edition *Descriptions Table*. This *Subset* was first available with the *SNOMED CT* Spanish Edition April 2002 *Release*.
German *Subset*	*Language*	Identifies *Descriptions* that comprise the German Edition of *SNOMED CT*. This *Subset* is required to determine the contents of the German release, which includes *descriptions* both in English and German. This *subset* applies to the German Edition *Descriptions Table*. This *Subset* was first available with the *SNOMED CT* German Edition April 2003 *Release*. It is not currently maintained or distributed.
SNOMED CT Top Level *Navigation Hierarchy*	Navigational *Subset*	Provides an example of a *Navigation hierarchy* using *SNOMED CT*'s Top Level *Concepts*. Please refer to the *Technical Implementation Guide* for more information about how to define and use *Navigation Subsets*. This *Subset* was first available with the First *Release* Developer Toolkit.

Subset	**Type of *Subset***	***Subset* Contents**
CTV3 Navigation Hierarchy	Navigational *Subset*	Provides an example of a *Navigation hierarchy* using *CTV3*'s top level *concepts*. Please refer to the *Technical Implementation Guide* for more information about how to define and use *Navigation Subsets*. This *Subset* was first available with the First *Release* Developer Toolkit.
Duplicate Terms Subset	*Duplicate Terms*	Identifies *Duplicate Terms* and the favored *term*. Refer to the *Developer Toolkit* in this guide for more information. This *Subset* was first available with the July 2002 Developer Toolkit.
US Drug *Extension Subsets*	*Realm Concept* and *Realm Description Subsets*	Identifies *Concepts* and *Descriptions* that are currently members of the US Drug *Extension*. These *Subsets* were first available with the July 2002 *Release*.
SNOMED CT Relationship Range Subset	*Navigation Subset*	Identifies the allowable range for each *SNOMED CT* RelationshipType *concept*. For example, the RelationshipType "Procedure site' receives as values or alternatively has a range of *concepts* that are descended from Anatomical *concepts*.
SNOMED CT Relationship Domain Subset	*Navigation Subset*	Identifies the allowable domain to which each *SNOMED CT* RelationshipType can be applied. For example, the RelationshipType *concept* "Procedure site' can be applied *concepts* that are descended from "Procedure.'

3.2.5 Navigation hierarchies

3.2.5.1 Representation of navigational Concepts

Navigation Concepts are *subtype children* of the Special *Concept* "*Navigation concept*." They have no *subtypes* of their own.

The "*Navigation concept*" has a designated *ConceptId* that is documented in *Table 1*.

3.2.5.2 Representation of navigational links

Navigation links are represented as rows of a *Navigation Subset*. Several *Navigation Subsets* may be released, each representing an alternative hierarchical view of *SNOMED CT*. Each *Subset Member* identifies a parent *Concept* (MemberID), an *order* (*MemberStatus*) and a *child Concept* (LinkedID).

3.2.6 Cross Mapping

The *Cross Mapping* mechanism enables distribution of *Cross Maps* from *SNOMED CT* in a common structure.

Cross Mappings ensure that *SNOMED CT* can be used to effectively reference other terminologies and classifications. Each *cross map* matches *SNOMED CT concepts* with another coding scheme that is called the "*target scheme* ".

The *Cross Mapping* structure enables:

- Automatic mapping from one *SNOMED CT Concept* to a single appropriate matching code in the *Target Scheme*;
- Automatic mapping from one *SNOMED CT Concept* to a single collection of codes in a *Target Scheme* that together represent the same *Concept*;
- Manual choice from a set of options for mapping a *SNOMED CT Concept* to a *Target Scheme* with several possible ways of representing the same or similar *Concepts*.

The *Cross Mapping* structure does **not** enable:

- Mapping from *postcoordinated* collections of *SNOMED CT Concepts* to a single *Target Code* or a specific collection of *Target Codes* (e.g. mapping a combination of a disorder qualified by severity or a procedure qualified by urgency).
- Mapping from multiple fields in a patient record to a specific *Target Code* that represents a combination of characteristics (e.g. mapping a combination of a disorder, procedure and the age and sex of the patient to a single grouper code).

This structure is based on the practical experience of the *Cross Mapping* tables of *Clinical Terms Version 3* (*CTV3*), which is one of *SNOMED CT*'s source terminologies.

The Cross *Mapping Mechanism* function is designed to be flexible in its support of:

1. The number of source *concepts* mapped:
   - A *Target Scheme* may include codes for:
     - A single *SNOMED CT Concept*:
       - **Example:** A disorder *Concept* may map directly to an ICD classification code.
     - A single statement consisting of a *postcoordinated* set of *Concepts*:
       - **Example:** A procedure *Concept* qualified by an urgency *Concept* may map to a target procedure code.
     - Several separate statements each represented using *SNOMED CT Concepts*:
       - **Example:** A diagnosis *Concept* plus a procedure *Concept* may map to a grouper code.
     - One or more statements represented using *SNOMED CT Concepts* combined with additional information that is not represented as a *SNOMED CT Concept*:
       - **Example:** A disorder *Concept* plus the age and sex of the person may map to an ICD classification code.
   - *Current* support is for mapping from a single *SNOMED CT Concept*. Additional functions may be considered in the future.

2. The number of *Target Codes*:
   - A *Target Scheme* may represent a mapped *Concept* as:
     - a single code;
     - as a combination of codes;
     - a choice of several possible mappings consisting of one or more codes.
   - *Current* support is for a single *Target Code*, multiple *Target Codes*, and a choice of single or multiple *Target Codes*.

3. The accuracy of mapping:
   - A target mapping may represent a *Concept*:
     - precisely.

- imprecisely:
  - target more specific than source;
  - target less specific than source;
  - imprecise but neither more nor less specific.

- An indication of accuracy can be associated with each *Cross Map*.

Practical requirements for mapping from *SNOMED CT* to other terminologies or classifications (*Target Schemes*) are not limited to mapping single *Concepts* in *SNOMED CT* to single *Target Codes*. The more general problem is to map information in a patient record that has been encoded using *SNOMED CT Concepts* into an appropriate *Target Code* or set of *Target Codes*.

It will be impossible to map some *Concepts* to any *Target Codes* in some *Target Schemes*.

A *SNOMED CT Concept* may be unmappable for one of several *reasons*:

- It expresses a *concept* that is outside the domain covered by the *Target Scheme*
- It is insufficiently detailed to provide a meaningful *Cross Mapping*:
  - In this case a mappable code may be found by refining the *SNOMED CT Concept* (i.e. selecting an **|** is a **|** *descendant*).
- It is too detailed and information will be lost in the mapping:
  - If this loss of detail is unimportant with respect to the purpose of the mapping, it is possible to map the most proximal more generalized *Concept* that is mappable, or to create the map but indicate that the match is not 1:1.
- It is inappropriate to map the *concept*:
  - For example, disorders that do not apply to humans need not be mapped to an ICD-9 code.

3.2.6.1 Cross Mapping - Structure diagram

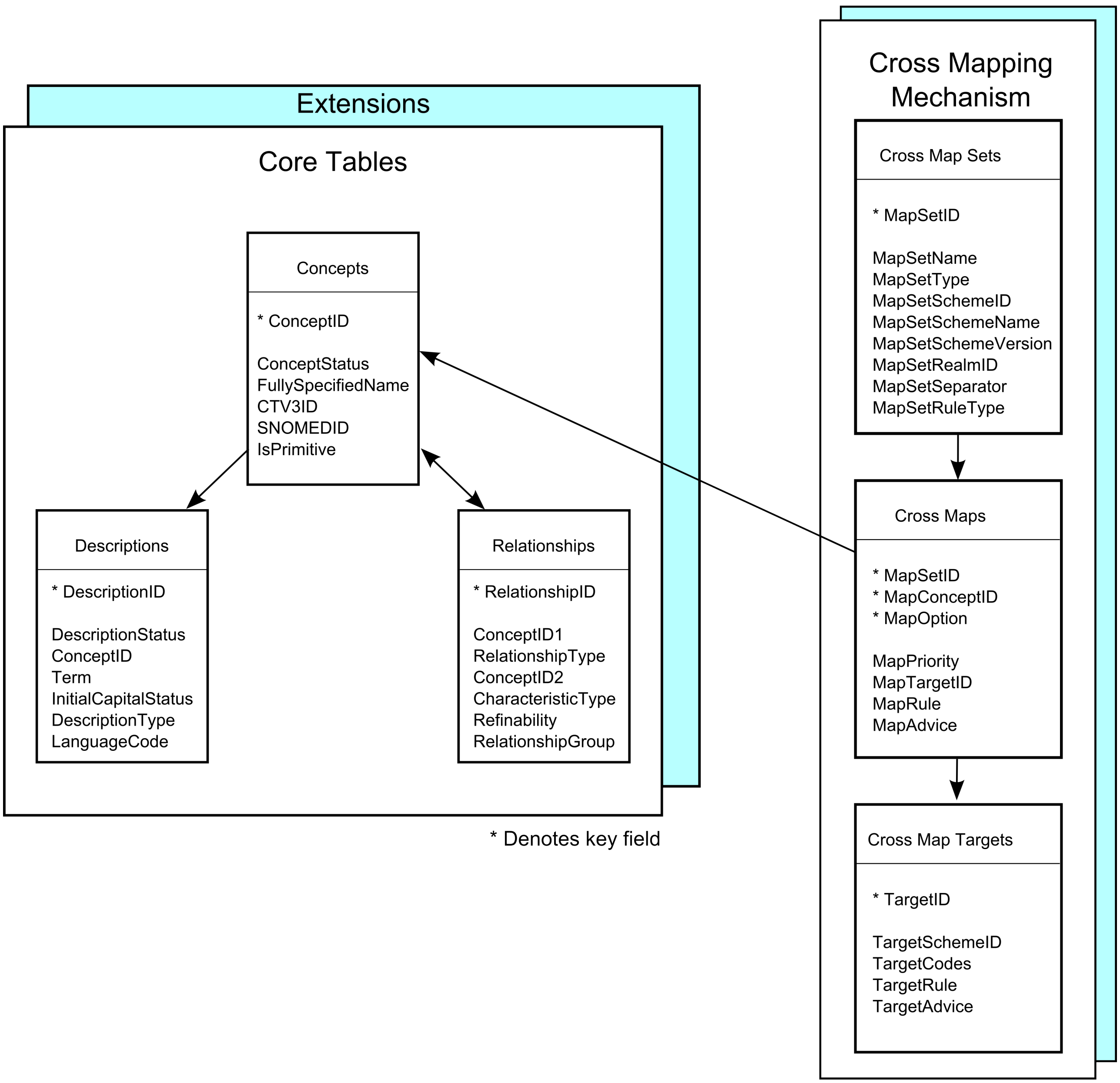


Figure 38: Cross Mapping Structure Overview (RF1)

- A *Cross Map Set* (*target scheme*) refers to 1-n *Cross Maps*;
- A *Concept* may be in 0-n *Cross Maps*;
- A *Cross Map* refers to 1 *Concept*;
- A Target can be used in 1-n *Cross Maps*;
- Each Target contains 1-n *target codes* (except for unmappable *Concepts*).

3.2.6.2 Cross Mapping Tables - Summary

The *SNOMED CT* structure for supporting *Cross Mapping* includes three tables:

Table 39: Cross Mapping Tables

Cross Map Sets Table	Each row in this table represents a *Target Scheme* for which *Cross Maps* are available.
Cross Maps Table	Each row in this table represents one option for mapping a *SNOMED CT Concept* to a *target code* or set of codes in the *Target Scheme*.
Cross Map Targets Table	Each row in this table represents a code or set of codes in the *Target Scheme*, which provides a mapping for one or more *SNOMED CT Concepts*.

The *relationships* of these tables to one another and to the *core tables* of *SNOMED CT* are described below. See *Release Format 1 - Detailed specification* on page 138 for details about the tables and *Cross Mapping Guide* on page 208 for information about available mappings.

3.2.6.2.1 Cross Maps Sets Table - Summary

Each row in the *Cross Map Sets Table* identifies a *Target Scheme* to which *SNOMED CT* is mapped and specifies characteristics of the associated mapping.

Table 40: Cross Map Sets Table

Key Fields	
MapSetId	The unique *SNOMED CT Identifier* for this *Cross Map Set*.

Data Fields	
MapSetName	A name that describes this *Cross Map Set*.
MapSetType	Indicates the nature of the *Cross Maps* associated with this scheme. CrossMapType is used to indicate the inclusion of one to one, one to many and choices of maps.
MapSetSchemeId	A standard *Identifier* for the *Target Scheme*.
MapSetSchemeName	The full name of the *Target Scheme*.
MapSetSchemeVersion	The version number of the *Target Scheme* as published by the issuing organization .
MapSetRealmId	The *Identifier* of the *Realm* within which this mapping table is applicable. This is only used in cases where *Realm* specific business rules or guidelines alter the acceptable mappings.
MapSetSeparator	The character used as a separator between the individual codes in the *Target Codes* field of the *Cross Map Targets*.

Data Fields	
MapSetRuleType	An indication of the types of rules used in the *Cross Maps* and *Cross Map Targets*.

3.2.6.2.2 Cross Maps Table - Summary

Each row in the *Cross Maps Table* represents one option for mapping a *Concept* to a *Cross Map Target*.

There may be several *Cross Map* options for a *Concept*. If so, each option is represented by a row in the *Cross Maps Table*. Each row may include rules for choosing that option and these rules may be expressed in machine-readable form and/or as textual advice to support manual coding.

Table 41: Cross Maps Table

Key Fields	
MapSetId	The unique *SNOMED CT Identifier* for the *Cross Map Set* of which this *Cross Map* is a member.
MapConceptId	The *SNOMED CT Identifier* of the mapped *Concept*.
MapOption	An *integer* that distinguishes between alternative mappings for a single *Concept*.

Data Fields	
MapPriority	Indication of the suggested *order* in which to present a series of options for mapping a *Concept* for manual assessment. The first of these is the default option for mapping the *Concept*.
MapTargetId	The unique *SNOMED CT Identifier* for a CrossMapTarget to which this *Concept* can be mapped.
MapRule	A machine processable *expression* of rules that determine whether this is an appropriate *Cross Map*.
MapAdvice	Textual advice to support manual mapping decisions between this *Cross Map* and other options for mapping the same *Concept*.

3.2.6.2.3 Cross Map Targets Table - Summary

Each row in this table represents a code or set of codes in the *Target Scheme*, which provides a mapping for one or more *SNOMED CT Concepts*. A *Cross Map Target* may include a rule indicating the conditions in which it applies; this feature is not used in any *Cross Map* currently released by the *IHTSDO*.

Table 42: Cross Map Targets Table

Key Fields	
TargetId	The unique *SNOMED CT Identifier* for this *Cross Map Target*

Data Fields	
TargetSchemeId	A standard *Identifier* for the *Target Scheme* from which the MapTargetCodes are derived
TargetCodes	A code or list of codes in the *Target Scheme* that represent an appropriate mapping for one or more *Concepts*
TargetRule	A machine processable *expression* of rules that determine the combinations of conditions to which this *Cross Map Target* applies
TargetAdvice	Textual advice expressing the combinations of conditions to which this *Cross Map Target* applies

3.2.6.2.4 SNOMED CT - LOINC® Integration Table - Summary

The Laboratory *LOINC*[8] database provides a standardized set of names and codes for identifying laboratory test results. Created by the Regenstrief Institute, the purpose of *LOINC* is to facilitate the exchange and pooling of results for outcomes management and research. Each *LOINC* code represents a unique laboratory test distinguished by six main parts that include the analyte measured, the property observed, the time aspect involved, the sample or system type, the scale of measurement, and where relevant the method of measurement. Taken together, these six components define a single laboratory test result. Additional details about *LOINC*, including access to the *LOINC* database files, is available from *www.regenstrief.org/loinc/loinc.htm*.

SNOMED CT is a clinical *reference terminology*. *SNOMED CT concepts* are explicitly represented in a multi-hierarchical structure. Each of the six distinguishing parts of a given *LOINC* test result are found in *SNOMED CT* and assigned in a *hierarchy*.

The integration exists as a table containing 11 columns. The first nine columns of the table are taken directly from the Laboratory *LOINC* version 1l database. The last two columns contain *concept identifiers* from *SNOMED CT* that relate a specific component of the *LOINC* test to the *SNOMED CT hierarchy*.

See *SNOMED CT - LOINC® Integration Table* for details of the *SNOMED CT - LOINC®* Integration Table.

3.2.6.3 Released Cross Mappings

The following *Cross Mappings* are currently included in the *International Release* available from the *IHTSDO*.

- ICD-9-CM (Clinical Findings);
- ICD-O (Version 2, Version 3);
- *LOINC* Integration Table.

[8] LOINC is a trademark of the Regenstrief Institute. Copyright 1995-2008, Regenstrief Institute and the Logical Observation Identifier Names and Codes (LOINC®) Committee. All rights reserved.

3.2.7 Content History

The content of *SNOMED CT* evolves with each release. The types of changes may include new *Concepts*, new *Descriptions*, and new *Relationships* between *Concepts*, new Cross Maps, new *Subsets*, as well as updates and retirement of any of these *components*. These changes are driven by changes in understanding of health and disease processes; introduction of new drugs, investigations, therapies and procedures; new threats to health; as well as proposals and work provided by *SNOMED CT* partners and licensees.

To achieve the goals stated in the widely-accepted paper on the desiderata for clinical terminologies, [9] the evolution of *SNOMED CT* follows these principles:

- Graceful evolution rather than radical change.
- The meaning of a *Concept* does not change. If the *Concept*'s meaning changes because it is found to be ambiguous, redundant or otherwise incorrect, the *Concept* is made *inactive*. One or more new *Concepts* are usually added to better represent the meaning of the old *Concept*.
- *Concepts* may become *inactive* but are never deleted. *concept identifiers* are persistent over time and are never reused.
- The link between a *Description* and a *Concept* is persistent. If a *Description* is no longer pertinent for a *Concept*, the *Description* is inactivated. A new corrected *Description* for the *Concept* may be added.
- Recognition of redundancy:
  - The same information can be stated two or more different ways. *SNOMED CT* is designed with a recognition of this possibility and facilitates recognition of equivalent statements.
  - When a *Concept* is inactivated as a result of duplication, *SNOMED CT* provides a reference to the continuing, *active* representation of that *Concept*.

A *SNOMED CT Component* is a *Concept*, *Description* or *Subset*. Changes to these components are included in the *Component History*. "Significant" changes generally require the component to be *retired* and replacement component(s) are added. The retirement and addition are recorded in the history records. Some changes have been designated as minor and only require a history record to record the change.

Why is this history important? These *components* may have been used directly in healthcare applications, or to define the *terms* used.

For example, an *inactive Description* or *Concept* may have been used in:

- Patient records;
- Data entry templates;
- Decision support protocols;
- Reusable searches, queries or analyzes

A change in a *Relationship* may alter the interpretation of stored information by protocols, queries and *Subsets* that include clauses which propagate across these *Relationships*. The history of *Relationship* changes may be computed by comparing different releases of the *Relationships* File; the *IHTSDO* does not provide this type of history file. Note that the *Relationships* File contains data about the history of the evolution of *Concepts* and supporting information.

Therefore, a *SNOMED CT enabled applications* may want to modify searches, software, or update data convention tables as the terminology evolves. The *SNOMED CT History Mechanism* provides not only a history of changes, but also references to the *current* components that may be used instead.

These references may be used in different ways.

- To map legacy data to a *current* equivalent *Concept*
- To enable a *query* or protocol to include information represented using *Inactive Concepts*

9 *Desiderata for Controlled Medical Vocabularies in the Twenty-First Century, J.J. Cimino, Methods of Information in Medicine 1998:37:394-403*

The *History Mechanism* is also used to track when the maintenance responsibility for one or more *concepts* is transferred between organizations . This requires the *concept* to be transferred from the *International Release* to an *Extension*, from an *Extension* to the *International Release*, or between *Extensions*. See *Rationale for Extensions* on page 23 for more information.

3.2.7.1 History Mechanism

The *history mechanism* involves these tables:

- *Component History Table*
- *Component History References Table*

All *SNOMED CT Components* have a unique *SNOMED CT Identifier*. The structure of these *Identifiers* is discussed in detail in *Component features - Identifiers* on page 20.

When a *component* is first added, it is allocated an *Identifier*, which is unique, permanent and cannot be changed. A *component* that is no longer required can be inactivated, and its *Identifier* will never be reused.

A limited set of permitted minor changes may be made to a *component* as specified by the *Component History* Rules.

A *component* may be replaced by another *component* to allow more significant changes. "Replace" means the original *component* is inactivated and one or more replacement *components* are added.

3.2.7.1.1 History Mechanism - Structure diagram

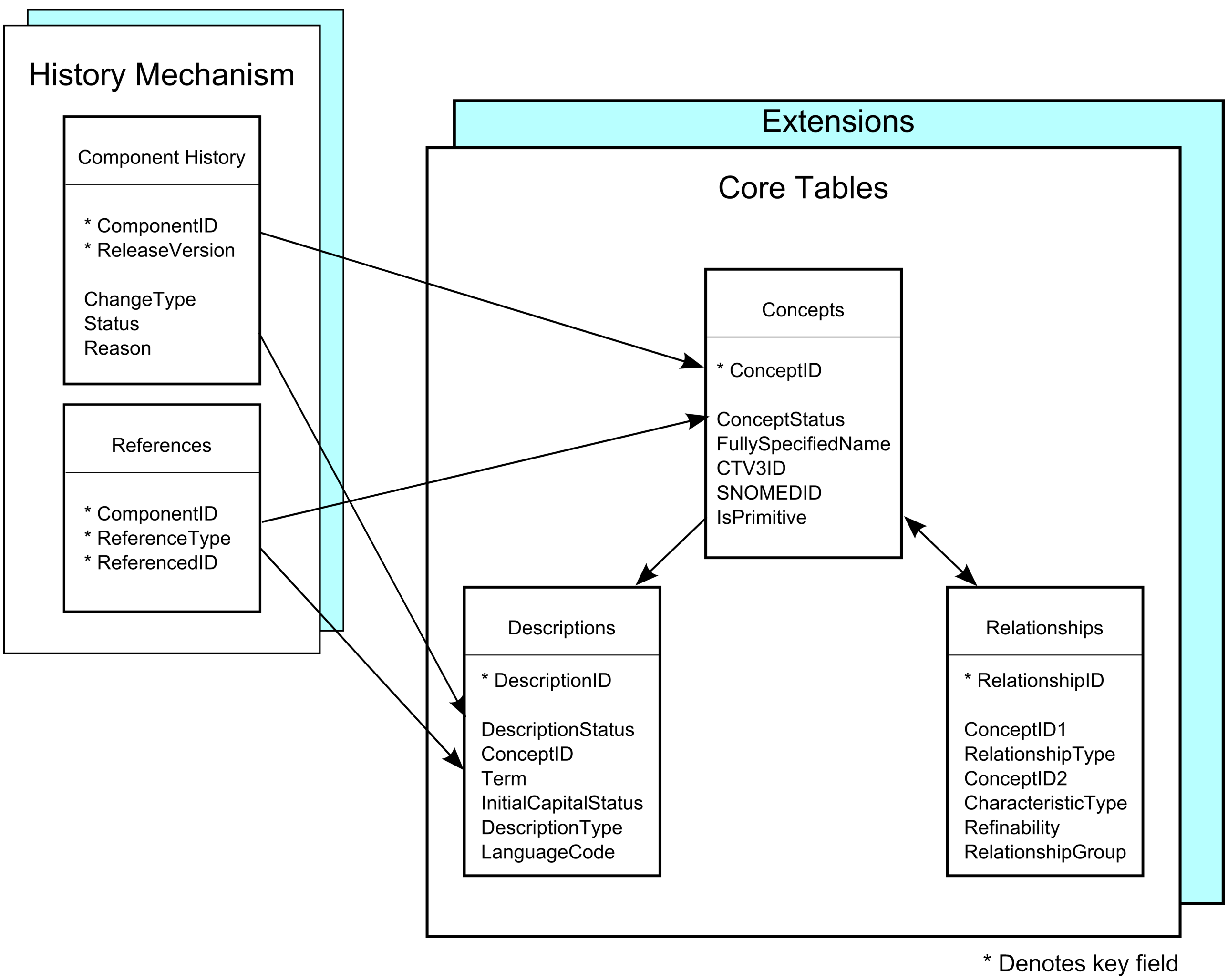


Figure 39: History Mechanism Overview (RF1)

A *Concept* or *Description* may be referenced in 0-1 *Component History* Rows per release.

A *Description* may be the ComponentID in 0-1 rows of the *References Table*.

A *Concept* or *Description* may be the ReferencedID in 0-n rows in the *References Table*.

3.2.7.1.2 Component History Table - Summary

The *Component History Table* identifies each change in the *status* of a *component* and the *Release Version* in which the change was made.

Table 43: Component History Table

Key Fields	
ComponentId	The unique *SNOMED CT Identifier* for the changed *component*
ReleaseVersion	The version of *SNOMED CT* in which this change was made

Data Fields	
ChangeType	An indication of the nature of the change
Status	The *status* of this *component* after the change
Reason	An optional text *Description* of the *reason* for the change

Note that the *Relationships* file shows the *Relationships* between *inactive concepts* to other equivalent or related *Concepts*.

3.2.7.1.3 References Table - Summary

The *References Table* contains References from *inactive components* to other equivalent or related *components* that were *current* in the *Release Version* in which that *component* was inactivated. Each Reference indicates the nature of the *relationship* between the *inactive* and persistent component.

The *References Table* was first distributed as part of the July 2008 *International Release* of *SNOMED CT*.

Table 44: References Table

Key Fields	
ComponentId	The unique *SNOMED CT Identifier* for the *inactive component*
ReferenceType	An indication of the nature of the *relationship* between the *inactive component* and the referenced *component*
ReferencedId	The unique *SNOMED CT Identifier* for the referenced *component*

With one exception, there are no *References Table* entries where *ComponentId* refers to a *Concept*; for most *Concepts*, this history tracking functionality is included in the *Relationships Table*. However, *ComponentId* may refer to an *inactive Concept* with *ConceptStatus* = 10 (*Moved Elsewhere*), in which case the value of ReferenceTypemust be "1" (Replaced By) or "4" (Alternative), and ReferencedIdwill refer to a *Concept*.

In addition, the *ReferencedId* field may also refer to a *Concept* when *ComponentId* refers to an *inactive Description* and *ReferenceType* = 7 (Refers to *Concept*). This use of the *References Table* permits a *Description* to be made *inactive* without having to retire its *Concept*. In such a case, the *References Table* indicates a *Concept* which is correctly described by the *Term* of the *inactive Description*.

3.2.7.1.4 Historical relationships

Historical relationships link *Inactive Concepts* to the overall semantic structure of *SNOMED CT*. These specialized *Relationships* should be used to map legacy data into the *current subtype hierarchy* of *SNOMED CT* :

- An "erroneous" *Concept* is related to the corrected *Concept* that replaces it by a | REPLACED BY |*Relationship*.
- A "duplicate" *Concept* is related to the *Concept* that it was found to duplicate by a | SAME AS |*Relationship*.
- An "ambiguous" *Concept* is related to the *Concept* (s) that represent its possible disambiguated meanings by one or more | MAY BE A | *Relationships*.
- A *Concept* that has the *status* "*Moved Elsewhere*" or "*Pending Move*" is related to a *Namespace Concept* by a | MOVED TO |*Relationship*. This identifies the *namespace* to which the *Concept* has been moved or will be moved.
- A *Concept* that replaces a *Concept* moved from another *namespace* has a | MOVED FROM |*Relationship* referring to the original *Concept* in its original *namespace*:
  - If the *Extension* from which a *Concept* originated is not installed the target *Concept* will not be available. This is not an error and can be safely ignored. The information in the *Relationship* is only of interest to users of the original *Concept*.
  - If the *Concept* has been moved to the *SNOMED CT core namespace* the | MOVED FROM |*Relationship* should be maintained in the originating *Extension*.

3.2.7.2 Component History Rules

These are the rules used to create the *Component History Table* when *components* are added or changed. *components* are never removed, but their *status* may change from *active* to *inactive*. *Components* that have become *inactive* may sometimes subsequently have their *status* restored to *active*.

Special considerations for the *SNOMED CT* First *Release* are described in the last subsection.

3.2.7.2.1 Adding a Component

A newly added *component* is distributed as:

1. A row in the *Component History Table* with:
   a. *ChangeType* value *Added*;
   b. ComponentIdreferring to the new *component*;
   c. *Status* value for the *component* (usually *Current*).

   .
2. A row in the appropriate distribution table for the new *component*.

3.2.7.2.2 Changing the status of a Component

All *SNOMED CT Components* have an *active* or *inactive Status*.

When the *Status* of a *component* is changed, this is distributed as:

1. A row in the *Component History Table* with:
   a. *ChangeType* value *Status Change*;
   b. *ComponentId* referring to the changed *component*;
   c. *Status* value of the *component* (usually one of the non-current *status* values).

   .

2. If the *component* is a *Concept* or *Description*, a row in the appropriate distribution table for the *component* with its updated *Status* (usually one of the non-current *status* values).

Components other than *Concepts* and *Descriptions* are only distributed when they are "*current*". Therefore, there is no *status* field in the distributed *Relationships*, *Subsets*, *Cross Map Sets* or *Cross Map Targets* Tables. Note that the *Relationships table* contains *Relationship Types* that describe *Relationships* between *inactive* and *active Concepts*.

Inactive Concepts and *Descriptions* are included in the distributed tables to support data recorded when these *components* were *active*. Therefore, the *status* of these *components* is represented explicitly in the *ConceptStatus* and *DescriptionStatus* fields of the distributed tables. The *reason* for inactivation of a *Concept* or *Description* may vary and is represented by a range of *status* values.

3.2.7.2.3 Making minor changes to Component

Most changes to a *component* require it to be inactivated and replaced by a new *component* with a new *SNOMED CT identifier*. However, a specified set of minor changes is permitted without inactivating the *component*.

A minor change to a *component* is distributed as:

1. A row in the *Component History Table* with:
   a. *ChangeType* value Minor Change;
   b. ComponentIdreferring to the changed *component*;
   c. *Status* value for the *component* (usually *Current*).

   .
2. A row in the appropriate distribution table for the changed *component*.

Minor changes to *components* are tabulated below:

Table 45: Permitted minor changes to components

Table	Field	Permitted minor changes
Concepts	FullySpecifiedName	Changes in presentation such as changed capitalization, punctuation, spelling or revision due to changes in agreed presentation style are permitted, without retiring the *ConceptId*, as long as the minor changes in FSN do not change the specified meaning of the *Concept*. Some changes to the semantic type shown in parentheses at the end of the FullySpecifiedName may also be considered minor changes if the change in hierarchy does not alter the *Concept*'s meaning. NOTE: This rule indicates that the *ConceptId* does not need to be retired in cases where the FSN undergoes minor changes. See below regarding required changes in *DescriptionId*.
Descriptions	*Term*	If *DescriptionType* is FullySpecifiedName then no changes are allowed; any change to any character of the FSN Term string requires a new *DescriptionId*. (NOTE: this is a new policy as of October 2011).
Descriptions	*Term*	If *DescriptionType* is not FullySpecifiedName Capitalization changes only.

Table	Field	Permitted minor changes
Descriptions	*DescriptionType*	If *DescriptionType* is not FullySpecifiedName Changes to any value except FullySpecifiedName.
Relationships	*Refinability*	Any change permitted.
Subsets	*SubsetName*	Changes that alter the presentation or more clearly indicate the purpose of a *Subset* are permitted.
Cross Map Sets	*MapSetName*	Changes that alter the presentation or more clearly indicate the purpose of a *Cross Map Set* are permitted.

3.2.7.2.4 Making significant changes to a Component

All changes other than those explicitly listed in the table of minor changes are regarded as significant changes.

A significant change to a *component* is represented as follows:

1. The *component* is inactivated. See *Component History Rules* on page 124 .
2. A new *component* with the changed information is added. See *Component History Rules* on page 124.
3. If the *component* is a *Description*, a *Subset* or a *Cross Map Set*, there will also be a row in the *Component History References Table* indicating the *Relationship* between the *inactive* and *active components* (Future Use).
4. If the *component* is a *Concept*, there will also be a row in the *Relationships Table* indicating the *relationship* between the *Inactive Concept* and its *active* replacement *Concept*.

3.2.7.2.5 Disambiguation of a Component

If a *Concept* is found to be ambiguous it may be replaced by one or more new *components*. These replacements are distributed in the same way as other significant changes. However, one or more new, *active components* are added to replace the *component* that was ambiguous.

3.2.7.2.6 Retiring, Inactivating, or removing a component

A *retired Concept* or *Description* is given an *inactive status*. The *inactive status* may describe the *reason* for the inactivation, such as duplicate. See *Component History Rules* on page 124. *Inactive Concepts* and *Descriptions* continue to be released with *SNOMED CT*.

A *Relationship* that is no longer valid is archived and no longer provided in the next *SNOMED CT release*. The *Relationship Identifier* will never be reused. Some " | is a |"*relationships* that are no longer valid may become "WAS A" *relationships* (Future Use).

3.2.7.2.7 Concept Status changes

This section is concerned with the way in which *Relationships* and *Descriptions* are affected by changes in the *status* of their associated *Concepts*. These changes are necessary because:

- An *Inactive Concept* cannot have any *active Descriptions*.
- There are *constraints* on the *Relationships* of an *Inactive Concept*. There are three distinct aspects.
- Changes to *Descriptions* - These changes are reflected in the *Descriptions Table*.
- Retirement, or replacement, of *Relationships*.
- Addition of *Relationships* that link an *Inactive Concept* to one or more *active Concepts* - These entries are added to the *Relationships Table*.

3.2.7.2.7.1 Impact on Descriptions

The table below shows the permitted DescriptionStatusvalues according to the ConceptStatusof the associated *Concept*.

Table 46: Permitted DescriptionStatus values for possible ConceptStatus values

Permitted *DescriptionStatus* values	**Permitted with *Concept Status***
0 - *Current* (default value)	0 - *Current*
6 - Limited (default value)	6 - Limited
8 - *Concept inactive* (default value)	1 - *Retired* without stated *reason* 2 - Duplicate 3 - Outdated 4 - Ambiguous 5 - Erroneous 10 - *Moved elsewhere*
1 - *Retired* without a stated *reason* 2 - Duplicate 3 - Outdated 5 - Erroneous 7 - Inappropriate	Any value
10 - *Moved elsewhere* (default value)	10 - *Moved elsewhere*
11 - *Pending move* (default value)	11 - *Pending move*

When the *Status* of a *Concept* changes, the *DescriptionStatus* of all associated *Descriptions* is checked:

- If it is in the list of permitted values for the new *ConceptStatus* it is not changed;
- If it is not in the list of permitted values for the new *ConceptStatus*, it is changed to the default value listed for that *ConceptStatus*.

Note that *Descriptions* that are already *inactive* at the time that a *Concept* is inactivated are left unchanged, while *active Descriptions* are changed to the value " *Concept* inactivated". Thus *Descriptions* that were *current* at the time a *Concept* is inactivated have *DescriptionStatus* = 8. However, valid *Descriptions* of a limited *Concept* remain are marked as limited (*DescriptionStatus* = 6).

3.2.7.2.7.2 Impact on Relationships - General issues

When a *Concept* is inactivated its *Relationships* are reassessed to avoid anomalies in the structures of *SNOMED CT*. The required and permitted *Relationships* following a change in *ConceptStatus* of one of the *Concepts* involved in the *Relationship* are summarized below.

Table 47: Relationships depending on Concept Status

ConceptId1 Concept Status	***Relationship Type***	***ConceptId2 Concept Status***	***Constraints***
1 - *Retired* without stated *reason* 3 - Outdated 5 - Erroneous 6 - Limited	WAS A	0 - *Current* 1 - *Retired* without stated *reason* 3 - Outdated 4 - Ambiguous 5 - Erroneous	No *constraints*
2 - Duplicate	SAME AS	0 - *Current* 11 - *Pending move*	One and only one *Relationship* of this type is required.
6 - Limited	SAME AS	6 - Limited	These may be reciprocal *relationships*.
4 - Ambiguous	MAY BE A	0 - *Current* 11 - *Pending move*	One or more *Relationships* of this type are required.
5 - Erroneous	REPLACED BY	0 - *Current* 11 - *Pending move*	One and only one *Relationship* of this type is permitted (optional). Must have at least one WAS A or one REPLACED BY.
1 - *Retired* without stated *reason* 3 - Outdated	REPLACED BY	0 - *Current*	One and only one *Relationship* of this type is permitted (optional).
10 - *Moved elsewhere* 11 - *Pending move*	MOVED TO	0 - *Current*	One or more *Relationships* of this type are required and must refer to a *Namespace Concept*.
Any value	MOVED FROM	10 - *Moved elsewhere* 11 - *Pending move*	Any number of these *Relationships* may exist (it is possible identical *Concepts* from more than one source were moved to this *Namespace*). *ConceptId2* must be in a different *Namespace* from *ConceptId1*.

Table 48: Defining and qualifying Relationships depending on Concept Status

ConceptId1 Concept Status	***Relationship Type***	***ConceptId2 Concept Status***	***Constraints***
0 - *Current* 11 - *Pending move*	| is a |	0 - *Current* 11 - *Pending move*	No *constraints*
1 - *Retired* without stated *reason* 2 - Duplicate 3 - Outdated 4 - Ambiguous 5 - Erroneous 6 - Limited 10 - *Moved elsewhere*	| is a |	0 - *Current*	One and only one *Relationship* of this type is required and it must refer to the appropriate *subtype* of "*Inactive concept* ".
0 - *Current* 11 - *Pending move*	Other defining *Relationships*	0 - *Current* 11 - *Pending move*	Any value appropriate to the *Relationship Type*.
0 - *Current* 11 - *Pending move*	Qualifying *Relationships*	0 - *Current* 11 - *Pending move*	Any value appropriate to the *Relationship Type*.
1 - *Retired* without stated *reason* 2 - Duplicate 3 - Outdated 4 - Ambiguous 5 - Erroneous 6 - Limited 10 - *Moved elsewhere*	Other defining or qualifying *Relationships*	NONE PERMITTED	NONE PERMITTED

3.2.7.2.7.3 Impact on Relationships- ConceptId1 status change

When a *Concept* is changed from being *active* to *inactive*, all of its defining and qualifying *Relationships* are inactivated. The " | is a |"*subtype Relationships* associated with *Concepts* that are inactivated without a stated *reason* or are stated to be outdated or erroneous may be converted to "WAS A"*relationships* to retain any significant semantics related to legacy data encoded using these *Concepts*.

The effects of this type of change may be wide reaching and have a significant impact on data files as well as software. Please refer to the *Technical Implementation Guide* for more information.

The full set of changes is summarized in the following table.

Table 49: Changes to Relationships when the status of ConceptId1 changes

***ConceptStatus*: Before**	***ConceptStatus*: After**	***Relationships* in which this is *ConceptId1***			
		| is a | *subtype*	**Other Defining**	***Qualifiers***	***Relationships* Added**
0	1	Retire or change to WAS A	Retire	Retire	| is a | "*Retired Concept*" (1) REPLACED BY (0 or 1)
	2	Retire	Retire	Retire	| is a | "Duplicate *Concept*" (1) SAME AS (1)
	3	Retire or change to WAS A	Retire	Retire	| is a | "Outdated *Concept*" (1) REPLACED BY (0 or 1)
	4	Retire	Retire	Retire	| is a | "Ambiguous *Concept*" (1) MAY BE A (1+)
	5	Retire or change to WAS A	Retire	Retire	| is a | "Erroneous *Concept*" (1) REPLACED BY (0 or 1)
	6	Retire or change to WAS A	Retire	Retire	| is a | "Limited *Concept*" (1) REPLACED BY (0 or 1)
0	10	Retire	Retire	Retire	| is a | "*Moved elsewhere Concept*" MOVED TO *Namespace* (1+)
0	11	No change	No change	No change	| is a | "*Pending move Concept*" MOVED TO *Namespace* (1+)

***ConceptStatus*: Before**	***ConceptStatus*: After**	***Relationships* in which this is *ConceptId1***			
		| is a | *subtype*	**Other Defining**	***Qualifiers***	***Relationships* Added**
11	10	Retire	Retire	Retire	| is a | "*Moved elsewhere Concept*" MOVED TO *Namespace* (unchanged)
11	0	No change	No change	No change	Retire: | is a | "*Pending move Concept*" Retire: MOVED TO *Namespace*
1-6	0	Retire	Not applicable	Not applicable	Defining *Relationships* and *qualifiers* as appropriate

3.2.7.2.7.4 Impact on Relationships- ConceptId2 changes

When a *Concept* is changed from being *active* to *inactive*, all the *Relationships* for which it provides the value (*ConceptId2*) must also be inactivated or replaced. In the case of a duplicate *Concept*, the value is replaced by a new similar *Relationship* with the *current Concept* that has the same meaning. In the case of an ambiguous *Concept*, the value may be replaced by a new similar *Relationship* with one of the disambiguated *Concepts* referred to by the ambiguous *Concept*. All *inactive concepts* can be found under the "Special *Concept*"*hierarchy*.

The effects of this type of change may be wide reaching and have a significant impact on data files as well as software. Please refer to the *Technical Implementation Guide* for more information.

Changes in *status* from *current* (*ConceptStatus* = 0) to limited (*ConceptStatus* = 6) constitute inactivation. A limited *Concept* cannot be the defining or qualifying value for any *Concept*. Any such *Relationships* are inactivated.

Note: Before January 2010 components with limited status (*ConceptStatus* = 6) were regarded as active. Therefore, when working with historic data use of components with this status should be regarded as use of an active component.

The full set of such changes is summarized in the following table.

Table 50: Changes to Relationships when the status of ConceptId2 changes

***ConceptStatus*: Before**	***ConceptStatus*: After**	***Status* of Concept1**	**| is a | *Relationships* in which this *Concept* is *ConceptId2***	**Other *Relationships* for which this *Concept* is *ConceptId2***
0 or 11	1	0 or 11	Retire	Retire
		1 to 6 or 10	Retire or change to WAS A	Retire
	2	Any	Retire and replace with equivalent *relationship* to the duplicated *Concept*.	Retire and replace with equivalent *relationship* to the duplicated *Concept*
	3	0 or 11	Retire	Retire
		1 to 6 or 10	Retire or change to WAS A	Retire
	4	Any	Retire and optionally replace with one of the disambiguated *Concepts*.	Retire and optionally replace with one of the disambiguated *Concepts*.
	5	Any	Retire or change to WAS A	Retire
	6	Any	Retire or change to WAS A	Retire
	10	Any	Retire and optionally replace by the *Concept* in the new target *namespace*.	Retire and optionally replace by the *Concept* in the new target *namespace*.
0	11	Any	No change	No change
11	0	Any	No change	No change

3.2.7.2.7.5 Changes to Relationships when status of the RelationshipType Concept changes

If a *Concept* that provides the RelationshipType for any *Relationships* is inactivated, all *Relationships* that refer to that *Concept* are inactivated and replaced.

3.2.7.2.7.6 Impact on Data Retrieval - Inactivation due to duplication

When a *Concept* is inactivated because it is found to be a duplicate of another *Concept* a new "SAME-AS"*Relationship* is created to link the *Inactive Concept* to the duplicate *Concept*.

This *Relationship* can be:

- Selectively included or excluded in data retrieval criteria to enable access to relevant information recorded using the *inactive* equivalent *ConceptId*;
- Used to allow legacy data to be upgraded to include the retained equivalent *ConceptId*.

3.2.7.2.7.7 Impact on Data Retrieval - Inactivation due to error

When a *Concept* is inactivated and replaced because it requires a significant change a new "REPLACED-BY"*Relationship* is created to link the *Inactive Concept* to the replacement *Concept*.

This *Relationship* can be:

- Selectively included or excluded in data retrieval criteria to enable access to relevant information recorded using the replaced version of the *ConceptId*;
- Used to allow legacy data to be upgraded to include the replacement *ConceptId*.

3.2.7.2.7.8 Impact on Data Retrieval - Inactivation due to ambiguity

When a *Concept* is inactivated and replaced because it was found to be ambiguous, new "MAYBE-A"*Relationships* are created to link the *Inactive Concept* to the *Concepts* that represent each of its possible meanings.

This *Relationship* can be:

- Selectively included or excluded in data retrieval criteria to enable access to relevant information recorded using the ambiguous *ConceptId*;
- If these *Relationships* are included the ambiguous *Concept* can be included in the results of a search for either of the possible meanings;
- Used to allow ambiguous data to be linked to the *ConceptIds* associated with either or both potential meanings.

3.2.7.2.7.9 Impact on Data Retrieval - Moves between namespaces

When a *Concept* is moved to another *Namespace*, a new | MOVED TO |*Relationship* is created pointing from the original *Concept* to the target *Namespace Concept*.

When a *Concept* is moved in from another *Namespace*, a new | MOVED FROM |*Relationship* is created pointing from the new *Concept* to the original *Concept* in the other *Namespace*.

When a *Concept* is moved in from an *Extension Namespace* into the *SNOMED CT Core Namespace*, the | MOVED FROM |*Relationship* is stored in the *Extension relationships table* and not the *SNOMED CT Core relationships table*.

These *Relationships* can be used to allow a check for the replacement *Concept* in the new target *Namespace*.

- The | MOVED TO | *Relationship* identifies the target *Namespace* to be searched;
- The | MOVED FROM | *Relationship*, in material distributed by the organization responsible for the target *Namespace*, allows identification of the replacement *Concept*.

The | MOVED TO | *Relationships* exist for *Concepts* with the ConceptStatusvalues *Moved Elsewhere* (10) and *Pending Move* (11). The difference between these *Status* values is significant.

The original *Concept* can still be used, retrieved and processed pending the availability of a replacement in the other *Namespace*. In some cases, a *Concept* marked as Pending Movemay be rejected for inclusion in the other *Namespace* and its *Status* in the original *Namespace* may revert to "*Current*".

3.2.7.2.7.9.1 Moved Elsewhere

The original organization is no longer supporting this as an *Active Concept*.

- It has no *Active Descriptions*
- The only *Relationships* that it has are:
    - " | is a |" "*Moved elsewhere Concept*".
    - | MOVED TO |:

- Therefore, without the replacement from the other *Namespace* this *Concept* cannot be processed or retrieved in any meaningful way.

3.2.7.2.7.9.2 Pending Move

The original organization is still supporting this as an *Active Concept*.

- Its *Descriptions* remain *active* (also marked with the same *Status* value).
- Its *Relationships* include:
  - Its pre-existing *Relationships* (or those deemed to be appropriate);
  - " | is a |" "*Pending move Concept*";
  - | MOVED TO |

3.2.7.3 Representation of material in the SNOMED CT First Release

This section details the initial settings in the *Component History Table* for the *SNOMED CT* First *Release*. These settings apply to *components* that were in *SNOMED RT* or Clinical *Terms* Version 3prior to the merger of these two terminologies. They also apply to *components* added to *SNOMED CT* prior to the First *Release*.

All *Concepts* in the First *Release* were regarded as having been added to *SNOMED CT* on or before the first release date (31 January 2002). Thus each has row in the *Component History Table* with *ChangeType* ="Added" and *ReleaseVersion* determined as follows:

- *Concepts* in releases of *SNOMED RT* (including those also present in *CTV3*):
  - *ReleaseVersion* = YYYYMMDD (*ISO* format of date previously recorded in RT *release files*).
- *Concepts* released in *Clinical Terms Version 3* before 31 January 2002 (excluding those also present in RT):
  - *ReleaseVersion* = "20020129".
- New *Concepts* in *SNOMED CT* First *Release* but not in either *SNOMED RT* or *Clinical Terms Version 3* before 31 January 2002:
  - *ReleaseVersion* = "20020131".
- New *Concepts* in any subsequent release of *SNOMED CT* (whether or not released in *CTV3* between *SNOMED CT releases*):
  - *ReleaseVersion* = YYYYMMDD (*ISO* date of the *SNOMED CT release*).

Some *Concepts* had a ConceptStatusother than *current* in the *SNOMED CT* First *Release*. This occurred if the *Concept*:

- was non-*current* in the source terminology

  or

  ;
- was rendered non-*current* during the merge of the two terminologies.

In either case there will be a row in *Component History Table* with the values *ChangeType* = " *Status* change", *Status* = *ConceptStatus* (with a *status* consistent with the First *Release*) and *ReleaseVersion* determined as follows:

- *Concepts* released in *Clinical Terms Version 3* before 31 January 2002 (excluding those also present in *SNOMED RT*) which were not *current* in *Clinical Terms Version 3* and were thus released in a non-*current* state in *SNOMED CT*:

    - *ReleaseVersion* = "20020130".

- *Concepts* released in *SNOMED RT* before 31 January 2002 which were *retired* in *SNOMED RT* and are thus released in a non-*current* state in *SNOMED CT*:

    - *ReleaseVersion* = YYYYMMDD (*ISO* format of date of change recorded in the RT *release files*).

- *Concepts* released in *CTV3* or *SNOMED RT* before 31 January 2002 that were rendered *inactive* prior to release of *SNOMED CT*:

    - *ReleaseVersion* = "20020131".

The consequences of these rules are shown below.

Table 51: New Concepts added in SNOMED CT with current status

ComponentId	***ReleaseVersion***	***ChangeType***	***Status***
ConceptId	20020131	0 (added)	0 (*current*)

Table 52: New Concepts added in SNOMED CT with limited status

ComponentId	***ReleaseVersion***	***ChangeType***	***Status***
ConceptId	20020131	0 (added)	6 (limited)

Table 53: SNOMED RT -sourced (or dual sourced) Concepts that were current in SNOMED CT First Release.

ComponentId	***ReleaseVersion***	***ChangeType***	***Status***
ConceptId	RT-release-date	0 (added)	0 (*current*)

***SNOMED RT* -sourced (or dual sourced) *Concepts* that were *current* in RT prior to 31 January 2002 but were non-*current* in *SNOMED CT* First *Release*.**

Table 54: SNOMED RT -sourced (or dual sourced) Concepts that were current in RT prior to 31 January 2002 but were non-current in SNOMED CT First Release.

The date 31 January 2002 indicates retirement occurred in First *Release*.

ComponentId	***ReleaseVersion***	***ChangeType***	***Status***
ConceptId	RT-release-date	0 (added)	0 (*current*)
ConceptId	20020131	1 (*status* change)	*ConceptStatus*

Table 55: SNOMED RT -sourced (or dual sourced) Concepts that were retired in RT prior to 31 January 2002 and were non- current in SNOMED CT First Release.

ComponentId	***ReleaseVersion***	***ChangeType***	***Status***
ConceptId	RT-release-date	0 (added)	0 (*current*)
ConceptId	RT- *retired* -date	1 (*status* change)	*ConceptStatus*

Table 56: Clinical Terms Version 3 -sourced Concepts that were current in SNOMED CT First Release.

ComponentId	***ReleaseVersion***	***ChangeType***	***Status***
ConceptId	20020129	0 (added)	0 (*current*)

Table 57: Clinical Terms Version 3 -sourced Concepts that were current in CTV3 prior to January 2002 but were non- current in SNOMED CT First Release.

The date 31 January 2002 indicates retirement occurred in first release.

ComponentId	***ReleaseVersion***	***ChangeType***	***Status***
ConceptId	20020129	0 (added)	0 (*current*)
ConceptId	20020131	1 (*status* change)	*ConceptStatus*

Table 58: Clinical Terms Version 3 -sourced Concepts that were not current in CTV3 prior to January 2002 and were non- current in SNOMED CT First Release.

The date 30 January 2002 indicates retirement prior to first release.

ComponentId	***ReleaseVersion***	***ChangeType***	***Status***
ConceptId	20020129	0 (added)	0 (*current*)
ConceptId	20020130	1 (*status* change)	*ConceptStatus*

Descriptions, *Relationships*, *Cross Maps* and *Subsets* in the First *Release* are treated as having been added as at 20020131 (31 January 2002). No history information is provided for these *components* prior to the First *Release*.

Note: The use of the dates 29, 30 and 31 of January 2002 is somewhat arbitrary. However, the chronological *order* of these dates is significant as it ensures that if *Component History Table* rows are applied in *ReleaseVersion* (i.e. date) *order*, the most recent change will be applied last. It also allows a distinction between items rendered *inactive* by the process of developing the *SNOMED CT* First *Release* and those that were already non-current in the source terminologies.

3.2.8 Distribution Files

3.2.8.1 Distribution Media

The *SNOMED CT International Release* is distributed to most Affiliates by an *IHTSDO Member National Release* Center . The *core files* are in the standard flat file structure for healthcare terminology. See the readme.txt file for a complete list of files included in the release. These files can be loaded into the specific file or database system used at your location.

3.2.8.2 File Formats

SNOMED CT files are *UTF-8* encoded to support character sets from around the world. Some database products, such as Oracle, require *UTF-8* to be specified in advance; if *UTF-8* is not specified, there may be some difficulty with using and displaying the special characters used in some of the *terms*. Note that these special characters are used occasionally in the *SNOMED CT* English *language* edition, as well as in non-English editions. Some English languagemedical *terms* are eponyms, based on the name of an individual, which may contain special characters.

The file format standard used starting in 2002 is now called "*Release Format* One" or *RF1*. There is now a draft format that will eventually replace it, known as "*Release Format* Two" or *RF2*. During the testing and transition phase, files in both formats may be available, and therefore it is important to have a file naming convention that distinguished them. This is done using the number 1 or 2 in the FileType Element. As an example, here is the file name of the *concepts* file for the January 2010 release, in the *current Release Format* (*RF1*):

- sct1_Concepts_Core_INT_20100131.txt.

The various parts of the *concepts table* name are given in the table below:

Table 59: File Naming Elements

Element	Value	Explanation
FileType	sct1	A terminology data file in *Release Format 1*
ContentType	*Concepts*	The data rows each represent a *concept*
ContentSubType	Core	The file is part of the *International Release* (*Release Format 1*)
Country or *Namespace*	INT	The file is part of the *International Release*
Version Date	20100131	an 8-digit number in the pattern "YYYYMMDD", in compliance with the *ISO* 8601 standard
Extension	txt	The file is in text format

A readme.txt file is distributed with each release. It contains a full list of all files and file sizes.

3.2.8.3 Directory Structures and Naming Conventions

Where compressed files are used for distribution, each archive may contain multiple files and subdirectories. The archives are named to reflect the types of files they contain, the YYYYMMDD version, and the file *extension* ".zip". For example, SNOMED_CT_International_Release_20070731_essential.zip was the name for the distributed zip archive for the essential files in the July 2007 release. Note that *National Releases* of *SNOMED CT* distributed by Member *National Release* Centers may use different directory structures and contain additional files.

3.2.8.4 Terminology version information

- Information about the *current* release is contained in a *Description* for the *Root Concept* Code.
- The *Description* carrying this information is a *Synonym* (*DescriptionType* =2) and its *status* is *current* (*DescriptionStatus* =0).
- The *Description* conveying version information for each release has a unique *DescriptionId* (i.e. it is **not** a revision of a previous versioning *Description*).
- In each release there is only one *Preferred Term* and one *current*"version" *Synonym* associated with the *Root Concept*. All previous version *Synonyms* are non- *current* (*DescriptionStatus* =1).
- The information in the "version" *synonym* is represented in the *Term* field as follows:
    - *SNOMED Clinical Terms* version: YYYYMMDD [*status*] (*description*):
        - *status* represents a word indicating of whether this is a release [R] or has some other *status* (e.g. for development set [D], evaluation [E], etc.).
        - yyyymmdd represents the release date in *ISO* format.
        - *description* is a textual *description* of the release (optional).
        - For example: *SNOMED CT* 20020131 [R].
- Other *synonyms* may be *active* for the *Root Concept* Code.

SNOMED CT enabled applicationsmust make this version information *synonym* visible, along with copyright and other similar information about the terminology.

3.2.9 Release Format 1 - Detailed specification

3.2.9.1 Concepts Table

Each row in the *Concepts Table* represents a clinical *concept*. Each *Concept* has a unique *Identifier* of the *concept* and a *Fully Specified Name* specifying the nature of the *concept*.

3.2.9.1.1 Key Fields

Table 60: -

Concepts Table			
Key Fields	**Field Type**	**Permitted characters**	**Length**
ConceptId	*SCTID*	Digits 0 to 9 only	6 to 18

The unique *SNOMED CT Identifier* for this clinical *concept*. See *Component features - Identifiers* on page 20 for an explanation of the *SCTID* data type format.

3.2.9.1.2 Data Fields

The *Concepts table* contains the following data fields:

- *ConceptStatus*
- FullySpecifiedName;
- *CTV3ID*
- *SNOMEDID*
- *IsPrimitive*

3.2.9.1.2.1 ConceptStatus

Table 61: ConceptStatus

Concepts Table			
Data Field	**Field Type**	**Permitted characters**	**Length**
ConceptStatus	Enumerated	See listed values	1 or 2

The *status* of a *Concept* indicates whether it is in *active* use and, if not, indicates the *reason* for withdrawal from *current* use.

Table 62: ConceptStatus Values

Value		**Meaning**
0	*Current*	The *Concept* is in *current* use and is considered *active*.
1	*Retired*	The *Concept* has been withdrawn without a specified *reason*. These *concepts* are considered *inactive*.
2	Duplicate	The *Concept* has been withdrawn from *current* use because it duplicates another *Concept*. These *concepts* are considered *inactive*.
3	Outdated	The *Concept* has been withdrawn from *current* use because it is no longer recognized as a valid clinical *concept*. These *concepts* are considered *inactive*.
4	Ambiguous	The *Concept* has been withdrawn from *current* use because it is inherently ambiguous. These *concepts* are considered *inactive*.
5	Erroneous	The *Concept* has been withdrawn from *current* use as it contains an error. A corrected but otherwise similar *Concept* may have been added to replace it. These *concepts* are considered *inactive*.
6	Limited	The *Concept* is of limited clinical value as it is based on a classification *concept* or an administrative definition. *Concepts* with this *status* are not valid for *current* use and are considered *inactive*.

Value		Meaning
10	*Moved elsewhere*	The *Concept* has been moved to an *extension*, to a different *extension*, or to the *International Release*. Use the |Moved To |*Relationship* to locate the *namespace* to which the *concept* has been moved. These *concepts* are considered *inactive*.
11	*Pending move*	The *Concept* will be moved to an *extension*, to a different *extension*, or to the *International Release*. Use the |Moved To |*Relationship* to locate the *namespace* to which the *concept* will be moved when the recipient organization confirms the move. These *concepts* are considered *active*.

3.2.9.1.2.2 FullySpecifiedName

Table 63: -

Concepts Table			
Data Field	**Field Type**	**Permitted characters**	**Length**
FullySpecifiedName	*String*	Any (except LF, CR and TAB)	1 to 255

A phrase that describes a *Concept* in a way that is intended to be unambiguous.

Each *Fully Specified Name* contains a suffix that indicates where it is integrated into the primary *hierarchy*. This can help distinguish, at a glance, for example, a finding from a disorder.

Table 64: Fully Specified Name Suffixes

• administrative *concept*	• assessment scale	• attribute
• body structure	• cell	• cell structure
• disorder	• environment / location	• environment
• ethnic group	• event	• finding
• geographic location	• *inactive concept*	• life style
• link assertion	• linkage *concept*	• morphologic abnormality
• *namespace concept*	• navigational *concept*	• observable entity
• occupation	• organism	• person
• physical force	• physical object	• procedure
• product	• *qualifier* value	• racial group
• record artifact	• regime/therapy	• religion/philosophy
• situation	• social *concept*	• special *concept*
• specimen	• staging scale	• substance
• tumor staging		

Note:

The *Fully Specified Name* is not the same as the *Preferred Term* in the *Descriptions Table*. The *Fully Specified Name* explains the meaning of the *concept* more fully than the *Preferred Term* to remove or reduce ambiguity.

The limitation in the length of this field is 255 bytes. All ASCII characters in the range32 to 127 can be encoded in a single byte using in *UTF-8*. However, accented and special characters require two or three byte encoding. The maximum length of a *Term* that contains these characters is less than 255 characters.

The *Fully Specified Name* is also represented as a row in the *Descriptions Table*. This allows translated versions of the *Fully Specified Name* to co-exist and to be referenced by appropriate *Language Subsets*. The *Fully Specified Name* in the *Concepts Table* is the American English *Fully Specified Name*.

Typically the *Fully Specified Name* will not be a *Term* that would be used in a clinical record. Instead, it may be stylized to ensure that it has a single unambiguous meaning.

Example: The *Preferred Term* of a *concept* could be "Aspiration of stomach contents', however, the *Fully Specified Name* might be "Aspiration of stomach contents (procedure)' to distinguish it from a post-operative complication.

3.2.9.1.2.3 CTV3ID

Table 65: CTV3ID

Concepts Table			
Data Field	**Field Type**	**Permitted characters**	**Length**
CTV3ID	*String*	0 to 9, A to Z, a to z or. (stop)	5

The *Read Code* for the *concept* taken from the United Kingdom's *Clinical Terms Version 3* terminology.

As *CTV3* is a superset of FourByte and Version 2 *Read Codes*, all the codes from those versions are present in *CTV3* and *SNOMED CT*. The *Identifiers* can be picked out easily as all FourByte codes have a leading "." and Version 2 codes have a trailing ".".

As new *concepts* are added to *SNOMED Clinical Terms*, new codes are used to populate this field. This will allow records to be maintained or searched using legacy codes during an extended *migration* period. The *IHTSDO* will review this commitment from time to time and it may be discontinued at some point in the future. In *Extensions*, these additional code fields may not be populated.

Note: The *CTV3ID* field should no longer be relied upon for mapping to and from the *Read Codes*. Additional mapping work in the UK identified some anomalies and resulted development of more flexibility table for *Read Code* Mapping

3.2.9.1.2.4 SNOMEDID

Table 66: -

Concepts Table			
Data Field	**Field Type**	**Permitted characters**	**Length**
SNOMEDID	*String*	0 to 9, A to Z or -(dash)	6 to 8

The legacy SNOMED International 3 to 3.5 code for the *Concept*..

As new *concepts* are added to *SNOMED Clinical Terms*, new codes compatible with legacy SNOMED International 3 to 3.5 were generated to populate this field. This will allow records to be maintained or searched using the legacy codes during an extended *migration* period. *IHTSDO* will review this commitment from time to time and it may be discontinued at some point in the future. In *Extensions*, this field may not be populated.

3.2.9.1.2.5 IsPrimitive

Table 67: -

Concepts Table			
Data Field	**Field Type**	**Permitted characters**	**Length**
IsPrimitive	*Boolean*	0 or 1	1

Indicates if a *Concept* is *Primitive* or *Fully defined* by its *defining characteristics*.

A *Concept* may be *Primitive* because:

- Its only stated *Defining characteristics* are " | is a |"*subtype Relationships*
- An aspect of its meaning is not fully expressed by existing *Relationships*.

If a *Concept* is *Primitive* its place in the *subtype hierarchy* may be known but it cannot be checked for *equivalence* with another *Concept expression* or *postcoordinated expression*.

Table 68: IsPrimitive Values

Values	
0	*Fully defined*
1	*Primitive*

3.2.9.2 Descriptions Table

Each row in the *Descriptions Table* associates a *term* with a clinical *concept*, which it can be used to represent. Each *Description* has its own unique *Identifier* and also contains the text of a *Term* and the *Identifier* of the *Concept* that it represents.

Note:

Some *terms* may be applied to more than one *concept*. In this case each is represented by a separate row in the *Descriptions Table* with the same *term* text but with different *Identifiers*.

Each *Description* contains a single *term*, which is valid in one or more *languages* or *dialects*.

To identify the *descriptions* for any *language* edition, the appropriate *Language Subset* must be used. Do no use the *Descriptions Table* alone. The *descriptions* for each *language* edition (including English) are designated in the *Subset*.

3.2.9.2.1 Key Fields

Table 69: -

Descriptions Table			
Key Fields	**Field Type**	**Permitted characters**	**Length**
DescriptionId	*SCTID*	Digits 0 to 9 only	6 to 18

The unique *SNOMED CT Identifier* for this *description*. See *Component features - Identifiers* on page 20 for an explanation of the *SCTID* data type format.

3.2.9.2.2 Data Fields

The *Descriptions Table* contains the following data fields:

- *DescriptionStatus*
- *ConceptId*
- *Term*
- *InitialCapitalStatus*
- *DescriptionType*
- *LanguageCode*

3.2.9.2.2.1 DescriptionStatus

Table 70: -

Descriptions Table			
Data Field	**Field Type**	**Permitted characters**	**Length**
DescriptionStatus	Enumerated	See listed values	1 to 2

The *status* of a *Description* indicates whether it is in *current* use and, if not, indicates the *reason* for withdrawal from *current* use.

Table 71: DescriptionStatus Values

Value		**Meaning**
0	*Current*	The *Description* and its associated *Concept* are in *current* use.
1	Non-*Current*	The *Description* has been withdrawn without a specified *reason*.
2	Duplicate	The *Description* has been withdrawn from *current* use because it duplicates another *description* containing the same *term* (or a very similar *term*) associated with the same *Concept*.

Value		Meaning
3	Outdated	The *Description* has been withdrawn from *current* use because this *Term* is no longer in general clinical use as a label for the associated *Concept*.
5	Erroneous	The *Description* has been withdrawn as the *Term* contains errors.
6	Limited	The *Description* is a valid *Description* of a *Concept* which has "limited" *status* (i.e. the *Concept* has *ConceptStatus* = 6).
7	Inappropriate	The *Description* has been withdrawn as the *Term* should not refer to this *concept*.
8	*Concept* non-*current*	The *Description* is a valid *Description* of a *Concept* which has been made non-*current* (i.e. the *Concept* has *ConceptStatus* 1, 2, 3, 4, 5, or 10).
10	*Moved elsewhere*	The *Description* has been moved to an *extension*, to a different *extension*, or to the *International Release*. A reference will indicate the *namespace* to which the *description* has been moved.
11	*Pending move*	The *Description* will be moved to an *extension*, to a different *extension*, or to the *International Release*. A reference will indicate the *namespace* to which the *description* has been moved when the recipient organization confirms the move (Future Use).

Note:

Status value 4 is only applicable to *ConceptStatus*.

A *Description* is only marked as *current* (*DescriptionStatus* = 0) if the associated *Concept* is also *current* (*ConceptStatus* = 0).

- A *Concept* made *inactive* (with *ConceptStatus* = 1, 2, 3, 4, 5 or 10) retains its previous *current Descriptions* but these are marked " *Concept* non-*current*" (*DescriptionStatus* = 8);
- An *inactive*"limited" *Concept* (*ConceptStatus* = 6) has valid usable *Descriptions* but these are also marked "limited" (*DescriptionStatus* = 6).

3.2.9.2.2.2 ConceptId

Table 72: -

Descriptions Table			
Data Field	**Field Type**	**Permitted characters**	**Length**
ConceptId	*SCTID*	Digits 0 to 9 only	6 to 18

The unique *SNOMED CT Identifier* of the clinical *concept* to which this *Description* applies.

Note:

This field provides the association between the *Descriptions* and *Concepts tables*.

3.2.9.2.2.3 Term

Table 73: -

Descriptions Table			
Data Field	**Field Type**	**Permitted characters**	**Length**
Term	*String*	Any (except LF, CR and TAB)	1 to 255

The text of a *Term* used to describe the associated *Concept*.

Note:

The *term* may be the FullySpecifiedName, the *Preferred Term* or a synonymous *term* for the associated *concept*. The *DescriptionType* field indicates the type of *term*. The *Preferred Term* is not the same as the *Fully Specified Name*. The *Preferred Term* expresses the *concept* accurately using *language* that a clinician would expect to find in a clinical record. In order to remove or reduce any possible ambiguity the *Fully Specified Name* may use phraseology that is stylized and unfamiliar.

The *Term* may be one or more *languages* or *dialects*. The *Descriptions* applicable to each *language* and/or *dialect* are indicated by inclusion in *Language Subsets* distributed with the *core tables*.

The limitation in the length of this field is 255 bytes. All ASCII characters in the range32 to 127 can be encoded in a single byte using *UTF-8*. However, accented and special characters require two or three byte encoding. Therefore, the maximum length for a *Term* that contains these characters is less than 255 characters.

The FullySpecifiedName is also present in the *Concepts Table*. It is included in the *Descriptions Table* to allow *language* and *dialect* variations to be represented.

3.2.9.2.2.4 InitialCapitalStatus

Table 74: -

Descriptions Table			
Data Field	**Field Type**	**Permitted characters**	**Length**
InitialCapitalStatus	*Boolean*	0 or 1	1

Indicates whether the capitalization *status* of the first character of the *Term* is significant.

Table 75: InitialCapitalStatus Values

Value		Meaning
0	False	The first character of the *Term* may be capitalized or uncapitalized according to its position in a sentence without changing its meaning. This is true for most *Terms*.
1	True	The capitalization of the first character of the *Term* must not be changed. This setting is used to indicate that the *Term* must retain an initial capital (e.g. "Down syndrome") or must not have its initial letter capitalized (e.g. "ml").

Note:

Capitalization of characters other than the first character in the *Term* is always regarded as significant.

3.2.9.2.2.5 DescriptionType

Table 76: -

Descriptions Table			
Data Field	**Field Type**	**Permitted characters**	**Length**
DescriptionType	Enumerated	See listed values	1

Indicates whether the *Term* is the *Preferred Term* or *Synonym* for the associated *Concept*.

Table 77: DescriptionType Values

Value		Meaning
0	Unspecified	This may be assigned as either a *Preferred Term* or *Synonym* by a *Description Subset* for a *language*, *dialect* or *realm*.
1	Preferred	This is the *Preferred Term* for the associated *Concept*.
2	*Synonym*	This is a *Synonym* for the associated *Concept*.
3	FullySpecifiedName	This is the FullySpecifiedName for the associated *Concept*.

Note:

The *Preferred Term* and *Synonyms* may vary according to *language* and *dialect*. These variations are specified using *Subsets*. In any one *language* or *dialect* there is one and only one *Preferred Term* for each *Concept*. The *Preferred Term* in one *language* or *dialect* may be a *Synonym* in other *languages* or *dialects*. Any *Description* with a *DescriptionType*"Unspecified", "Preferred" or "*Synonym*" in the distributed table may be assigned as a *Preferred Term* or *Synonym* in any *language* or *dialect*.

The FullySpecifiedName will vary according to *language* and *dialect*. These variations are specified using *Subsets*. In any *language* or *dialect* there is one FullySpecifiedName for each *Concept*. Only *Descriptions* that have the *DescriptionType*"FullySpecifiedName" may be assigned as the FullySpecifiedName in any *language* or *dialect*.

3.2.9.2.2.6 LanguageCode

Table 78: -

Descriptions Table			
Data Field	**Field Type**	**Permitted characters**	**Length**
LanguageCode	*String*	0 to 9, a to z, A to Z and dash "-"	1 to 8

This field specifies the *language*/*dialect* of the *term*.

A *string* identifying a *language* and, if appropriate, a *dialect* in which this *Description* is valid. Consists of a code and optionally a sub-code. If a sub-code is present it is separated from the code by a dash ("-").

- The code is the two-character ISO639-1 *language* code. ISO639 is the International Standard for “Codes for the representation of names of languages”.
- The sub code is a *string* of upper-case letters that represent the *dialect*. This deliberately mirrors the W3C approach and will either be:
  - If the *dialect* is general to an entire country, the two-letter *ISO* 3166 country code is used. ISO3166 is the International Standard for “Codes for the representation of names of countries”.
  - If *dialects* are used that are less common or not country or *language* linked, the IANA approach is used. This code consists of a *string* of more than two letters. IANA is the Internet Assigned Numbers Authority.

This structure follows Internet conventions. Examples: “en” for English, “es” for Spanish, “en-US” for United States English, “en-GB” for British English.

To identify the members of an edition of *SNOMED CT*, use the *Language subset* for that edition. Do not simply search for all entries in the *Descriptions Table* using the *Language* Code field.

Each edition includes the appropriate *Language Subset*, which designates the *descriptions* that are intended for that edition.

3.2.9.3 Relationships tables

Each row in the *Relationships Table* represents a *Relationship* between two *Concepts*.

Note: Most *Relationship Types* are directional and are therefore regarded as having source *concept* and a target *concept*. Reciprocal *relationships* are not explicitly represented by rows in the *Relationships Table*.

Example: If B is a *subtype* of A, it follows that A is a supertype of B. The first of these *Relationships* is represented by a row in the *relationships tables*. The reciprocal *Relationship* is implied and is not restated by another row in the table

Note:

The *Relationship Types* supported include the |is a| (*subtype*) *relationship*. This is a hierarchical *relationship* and only the closest *relationships* are represented explicitly. Other *relationships* are subsumed and are not represented by rows in the *Relationships Table*.

Example: If C is a *subtype* of B and B is a *subtype* of A, it follows that C is a *subtype* of A. The *Relationship* between C and A is not represented by a row in the *relationships tables* but is subsumed by the chain of *Relationships* between C and B and B and A.

Relationships other than | is a | (*subtype*) *relationship* are called attribute *relationships*. These *relationships* include the *concept* (object), *relationship type* (attribute), and *attribute value* (another *concept*). This is sometimes called the OAV triplet, which stands for Object-*Attribute*-Value.

Some sets of *relationships* may have an expected or logical display *order* (e.g. cranial nerves or cervical vertebrae). The *order* is described using the *subset* mechanism and is not part of the *Relationships Table*.

In 2003-2004, some *relationships* were packaged into a separate file called the Historical RelationshipsTable. These *relationships* were merged into the *Relationships Table* starting with the January 2005 release.

To recreate these two tables, use these SQL commands:

```
Create table historical_relationships as
Select * from  relationships
Where characteristic type = "2';
Delete from  relationships
Where characteristic type = "2';
```

Note:

The type of *relationship* is itself described using a *concept*.

3.2.9.3.1 Key Fields

Table 79: -

Relationships Table			
Key Fields	**Field Type**	**Permitted characters**	**Length**
RelationshipId	*SCTID*	Digits 0 to 9 only	6 to 18

The unique *SNOMED CT Identifier* for this *relationship*. See *Component features - Identifiers* on page 20 for an explanation of the *SCTID* data type format.

Note:

This field allows a *Relationship* to be referenced by a *Subset* or *Extension*.

3.2.9.3.2 Data Fields

The *Relationships Table* contains the following data fields:

- *ConceptId1*
- RelationshipType;
- *ConceptId2*
- *CharacteristicType*
- *Refinability*
- *RelationshipGroup*

3.2.9.3.2.1 ConceptId1

Table 80: -

Relationships Table			
Data Field	**Field Type**	**Permitted characters**	**Length**
ConceptId1	*SCTID*	Digits 0 to 9 only	6 to 18

The unique *SNOMED CT Identifier* of the *Concept* which is the source of this *Relationship*.

Note:

This field provides the first association between the *Relationships* and *Concepts tables* and is sometimes referred to as the "object."

3.2.9.3.2.2 RelationshipType

Table 81: -

Relationships Table			
Data Field	**Field Type**	**Permitted characters**	**Length**
RelationshipType	*SCTID*	Digits 0 to 9 only	6 to 18

The unique *SNOMED CT Identifier* of the *Concept* which represents the type of *relationship* between the related *Concepts*.

Note:

A special category of *Concepts* is used to represent RelationshipTypes.

This field provides the second association between the *Relationships* and *Concepts tables* and is sometimes called the " attribute" or "role."

3.2.9.3.2.3 ConceptId2

Table 82: -

Relationships Table			
Data Field	**Field Type**	**Permitted characters**	**Length**
ConceptId2	*SCTID*	Digits 0 to 9 only	6 to 18

The unique *SNOMED CT Identifier* of the *Concept* which is the target of this *Relationship*.

Note: This field provides the third association between the *Relationships* and *Concepts tables* and is sometimes called the "value."

3.2.9.3.2.4 CharacteristicType

Table 83: -

Relationships Table			
Data Field	**Field Type**	**Permitted characters**	**Length**
CharacteristicType	Enumerated	See listed values	1

An indication of whether a *Relationship* specifies a *defining characteristic* of the source *Concept* or a possible qualification of that *Concept*.

Note:

The *Relationships Table* is concerned with definitions, *qualifiers* and additional facts about a *Concept*. Although it can also be used to display a *hierarchy* of *subtypes* it does not have a specified or natural display order. More effective *browser navigation* is supported by use of a *Navigation Subset*.

Table 84: CharacteristicType Values

Value		**Meaning**
0	Defining	This *relationship* represents a *defining characteristic* of the source *concept*. Hierarchical *relationships* (e.g. | is a |) are also regarded as defining *relationships*. **Example:**" | procedure site | = | liver |" is a *defining characteristic* of "Liver biopsy'.
1	*Qualifier*	This *relationship* represents an optional *qualifying characteristic*. **Example:** " | revision status | = | first revision |" is a possible qualification of | hip replacement |.
2	Historical	This is used to relate an *inactive concept* to another *concept*. **Example:** The | SAME AS | *relationship* connects an *inactive concept* with the *concept* it duplicates.
3	Additional	This *relationship* represents a *Context specific characteristic*. This is used to convey characteristics of a *concept* that apply at a particular time within a particular organization but which are not intrinsic to the *concept*. **Example:** “Prescription Only Medicine” is a *Context specific characteristic* of the *Concept* | amoxycillin 250mg capsule |. It is true currently in the UK but is not true in some other countries.

3.2.9.3.2.5 Refinability

Table 85: -

Relationships Table			
Data Field	**Field Type**	**Permitted characters**	**Length**
Refinability	Enumerated	See listed values	1

An indication of whether it is possible to refine the target *concept* when this *Relationship* is used as a template for clinical data entry.

Table 86: Refinability Values

Value		**Meaning**
0	Not refinable	Not refinable.
1	Optional	May be refined by selecting *subtypes*.
2	Mandatory	Must be refined by selecting a *subtype*.

3.2.9.3.2.6 RelationshipGroup

Table 87: -

Relationships Table			
Data Field	**Field Type**	**Permitted characters**	**Length**
RelationshipGroup	*Integer*	0 to 9	1 to 2

An *integer* value that expresses an association between two or more *Relationships*.

The default *Relationship group* value is zero and this applies to all *Relationships* that have not been stated to be associated with any other *Relationships*. All *Relationships* that share the same *ConceptId1* and the same non-zero *Relationship group* value are associated with one another. Any *Relationships* that share the same *ConceptId1* but have different *Relationship group* values are not associated with one another. See example below.

Table 88: -

ConceptId1	**RelationshipType**	***ConceptId2***	**RoleGroup**
Ureteroscopic removal of ureteric calculus	Direct morphology	Calculus	1
Ureteroscopic removal of ureteric calculus	Method	Removal	1

ConceptId1	RelationshipType	*ConceptId2*	RoleGroup
Ureteroscopic removal of ureteric calculus	Procedure site - Indirect	Ureteric structure	1
Ureteroscopic removal of ureteric calculus	Method	Surgical action	2

3.2.9.4 Subset Mechanism

3.2.9.4.1 Subset Table

Each row in the *Subsets Table* represents a *Subset*. Each *Subset* has a unique *Identifier*, a *SubsetName*, a *SubsetType* and a collection of additional characteristics.

3.2.9.4.1.1 Key Fields

Table 89: -

Subsets Table			
Key Fields	**Field Type**	**Permitted characters**	**Length**
SubsetId	*SCTID*	Digits 0 to 9 only	6 to 18

The unique *SNOMED CT Identifier* for this *subset*. See *Component features - Identifiers* on page 20.

3.2.9.4.1.2 Data Fields

The *Subsets table* contains the following data fields:

- *SubsetOriginalId*
- SubsetVersion;
- *SubsetName*
- *SubsetType*
- *LanguageCode*
- *RealmId*
- *ContextId*

3.2.9.4.1.2.1 SubsetOriginalId

Table 90: -

Subsets Table			
Data Field	**Field Type**	**Permitted characters**	**Length**
SubsetOriginalId	*SCTID*	0 to 9	6 to 18

The unique *SNOMED CT Identifier* for the first version of the *Subset* on which this *Subset* is based.

Note:

For the first version of a *Subset* the *SubsetOriginalId* and *SubsetId* fields contain the same value. For each subsequent version the SubsetVersion is incremented and a new *SubsetId* is allocated but the *SubsetOriginalId* field retains the same value in all versions.

3.2.9.4.1.2.2 SubsetVersion

Table 91: -

Subsets Table			
Data Field	**Field Type**	**Permitted characters**	**Length**
SubsetVersion	*Integer*	0 to 9	1 to 5

An integerincreased for each revised release of this *Subset*.

Note:

A single distribution table will only contain one version of a *Subset*. However, legacy versions of a *Subset* may need to be retained to support other *Subset Definitions* or continued use of a protocol which has not been validated for use with the revised *Subset*.

3.2.9.4.1.2.3 SubsetName

Table 92: -

Subsets Table			
Data Field	**Field Type**	**Permitted characters**	**Length**
SubsetName	*String*	Any (except LF, CR and TAB)	1 to 255

A descriptive name given to the *Subset* by its originator.

3.2.9.4.1.2.4 SubsetType

Table 93: -

Subsets Table			
Data Field	**Field Type**	**Permitted characters**	**Length**
SubsetType	*Integer*	0 to 9	1 to 2

Indicates the nature of the *Subset* and the type of *SNOMED CT Component* that may be a member of the *Subset*.

Table 94: SubsetType Values

Value		Meaning
1	*Language*	*Descriptions* appropriate to a specified *Language* or *Dialect*.
2	*Realm Concept*	*Concepts* appropriate to a specified *Realm*.
3	*Realm Description*	*Descriptions* appropriate to a specified *Realm*.
4	*Realm Relationship*	*Relationships* appropriate to a specified *Realm* (for future use).
5	Context *Concept*	*Concepts* appropriate to a specified *Context Domain*.
6	Context *Description*	*Descriptions* appropriate to a specified *Context Domain*.
7	*Navigation*	*Navigation Links* that represent an ordered *Navigation hierarchy*.
8	*Duplicate Terms*	*Descriptions* that contain identical *Terms*. Other values are reserved for future use.

3.2.9.4.1.2.5 LanguageCode

Table 95: -

Subsets Table			
Data Field	**Field Type**	**Permitted characters**	**Length**
LanguageCode	*String*	0 to 9, a to z, A to Z and dash "-"	1 to 8

This field specifies the *language* / *dialect* of the *descriptions* in a *subset*.

A *string* identifying a *Language* and, if appropriate, a *dialect* to which the *Descriptions* in this *subset* are valid. Consists of a code and optionally a sub-code. If a sub-code is present it is separated from the code by a dash ("-").

- The code is the two-character ISO639-1 *language* code. ISO639 is the International Standard for "Codes for the representation of names of *languages* ".
- The sub code is a *string* of upper-case letters that represent the *dialect*. This deliberately mirrors the W3C approach and will either be:
    - If the *dialect* is general to an entire country, the two-letter *ISO* 3166 country code is used. ISO3166 is the International Standard for "Codes for the representation of names of countries".
    - If *dialects* are used that are less common or not country or *language* linked, the IANA approach is used. This code consists of a *string* of more than two letters. IANA is the Internet Assigned Numbers Authority.

This structure follows Internet conventions. Examples: "en" for "English", "es" for Spanish, "en-US" for United States English, "en-GB" for British English.

Usage: Required for *Language Subsets*, *Realm Description Subsets* and *Context Description Subsets*. In all other *SubsetTypes* the *LanguageCode* has the value = "0" (zero) and should be ignored.

To identify the members of an edition of *SNOMED CT*, use the *Language subset* for that edition. Do not simply search for all entries in the *Descriptions Table* using the *Language* Code field.

Each edition includes the appropriate *Language Subset*, which designates the *descriptions* that are intended for that edition.

3.2.9.4.1.2.6 RealmId

Table 96: -

Subsets Table			
Data Field	**Field Type**	**Permitted characters**	**Length**
RealmId	*String*	0 to 9 and full-stop "."	4 to 24

A stringidentifying a *Realm*.

For example, UK GP *subset* (a *realm subset* for general practice physicians in the United Kingdom).

Usage Required for *Realm Concept Subsets*, *Realm Description Subsets*, *Context Concept Subsets* and *Context Description Subsets*. May also be used in *Duplicate Terms Subsets*.

In all other *Subset Types* the *RealmId* has the value "0" (zero) and should be ignored.

3.2.9.4.1.2.7 ContextId

Table 97: -

Subsets Table			
Data Field	**Field Type**	**Permitted characters**	**Length**
ContextId	*String*	0 to, 9 a to z, A to Z, "." and "-"	1 to 18

A *string* that identifies a *Context Domain* within the *Realm* indicated by the *RealmId*. *Context Domain* may be specified for different purposes. These include:

- Categories of *Concepts* with particular roles in the terminology or in a record:
  - *Concepts* that have an organizing role in the terminology but which are not appropriate to use in an individual patient record (e.g. the *Root Concept*, Top-Level *Concepts* and other *Concepts* that provide a rational hierarchical structure without the precision appropriate to a patient record.
  - *Concepts* are applicable as organizing headers in a record but which are not appropriate for use in making clinical statements.
- Broad contexts for use within parts of a record:
  - *Concepts* that are appropriately detailed for use as a discharge diagnosis according to a local convention or agreed protocol.
- Narrow contexts used to constrain data entry or population of a message field:

- *Concepts* or *Descriptions* to be presented as valid choices for a particular field in a screen or protocol.
- *Concepts* that are applicable to a message field. Examples include:
  - An *HL7* "value domain";
  - The *NHS* Pathology Message bounded code list.

Usage: Required for *Context Concept Subsets* and *Context Description Subsets*. May also be used in *Duplicate Terms Subsets*.

In all other *Subset Types* the *ContextId* has the value "0" (zero) and should be ignored.

3.2.9.4.2 Subset Members Table

3.2.9.4.2.1 Subset Members Table: Common Details

Each row in the *Subset Members Table* represents one member of a *Subset*. The *Subset* to which a member belongs is identified by the *Subset* ID.

The member is identified by a *MemberId*, which refers to a *Concept*, *Description*, or *Relationship*.

The *MemberStatus* and *LinkedId* fields are interpreted depending on the *SubsetType*. Each *type of subset* is described separately. The types of *subsets* are:

- *Language Subset*
- *Realm Concept Subset*
- *Realm Description Subset*
- *Realm Relationship Subset* (for future use);
- *Context Concept Subset*
- *Context Description Subset*
- *Navigation Subset*
- *Duplicate Terms Subset*

The *MemberId* may be a *ConceptId*, *DescriptionId*, or *RelationshipId*.

3.2.9.4.2.1.1 Key Fields

The *Subset Members Table* has the following key fields:

- *SubsetId*
- *MemberId*
- *MemberStatus*

3.2.9.4.2.1.1.1 SubsetId

Table 98: -

***Subsets* Members Table**			
Key Fields	**Field Type**	**Permitted characters**	**Length**
SubsetId	*SCTID*	0 to 9	6 to 18

The unique *SNOMED CT Identifier* for the *Subset* to which this applies.

3.2.9.4.2.1.1.2 MemberId

Table 99: -

***Subsets* Members Table**			
Key Fields	**Field Type**	**Permitted characters**	**Length**
MemberId	*SCTID*	0 to 9	6 to 18

The *SNOMED CT Identifier* of a *Subset Member*. The permitted *Subset Members* depends on the *SubsetType* and can be a *ConceptId*, *DescriptionId*, or *RelationshipId*.

3.2.9.4.2.1.1.3 MemberStatus

Table 100: -

***Subsets* Members Table**			
Key Fields	**Field Type**	**Permitted characters**	**Length**
MemberStatus	*Integer*	0 to 9	1 to 5

The *status* of the identified member in this *Subset*.

The value of *MemberStatus* must be greater than zero.

The interpretation of MemberStatusvalues depends on the *SubsetType*.

For most *Subsets* the combination of *Subset* ID and *MemberId* is unique. The exception to this rule is *Navigation Subsets* and for this *reason* the *MemberStatus* is included in the key.

3.2.9.4.2.1.2 Data Fields

Table 101: -

Subsets Table			
Data Field	**Field Type**	**Permitted characters**	**Length**
LinkedId	*SCTID*	0 to 9, Null	6 to 18

The Interpretation of *LinkedId* depends on the *SubsetType* and may be null in some cases.

3.2.9.4.2.2 Language Subset Members Tables

3.2.9.4.2.2.1 Key Fields

The *Language Subset Members Table* has the following key fields:

- *SubsetId*
- *MemberId*
- *MemberStatus*

3.2.9.4.2.2.1.1 SubsetId

Table 102: -

Language Subset Members Table			
Key Fields	**Field Type**	**Permitted characters**	**Length**
SubsetId	*SCTID*	0 to 9	6 to 18

The unique *SNOMED CT Identifier* for the *Subset* to which this applies.

3.2.9.4.2.2.1.2 MemberId

Table 103: -

Language Subset Members Table			
Key Fields	**Field Type**	**Permitted characters**	**Length**
MemberId	*SCTID*	0 to 9	6 to 18

The *Identifier* of a *SNOMED CT Description*.

3.2.9.4.2.2.1.3 MemberStatus

Table 104: -

Language Subset Members Table			
Key Fields	**Field Type**	**Permitted characters**	**Length**
MemberStatus	*Integer*	0 to 9	1 to 5

The *status* of the identified member in this *Subset*.

The value of *MemberStatus* must be greater than zero.

Language Subset The *MemberStatus* specifies the role of the referenced *Description* in this *Language*. There are three roles.

1 = *Preferred Term*
2 = *Synonym*
3 = *Fully Specified Name*

Note: The following rules must apply to each *Subset*:

1. A *Description* cannot be assigned the role *Fully Specified Name* unless its *DescriptionType* is also "*Fully Specified Name* ";
2. Only one *Description* of each *Concept* may be assigned as the *Fully Specified Name*;
3. If no *Description* of a *Concept* is assigned the role *Fully Specified Name*, the *Fully Specified Name* in the *Concepts Table* assumes this role;

4. A *Description* cannot be assigned the role *Preferred Term* or *Synonym* if its *DescriptionType* is "*Fully Specified Name* ";
5. One and only one *Description* of each *Concept* must be assigned the role "*Preferred Term* ";
6. Any number of *Descriptions* of a *Concept* may be assigned the role "*Synonym* ".

3.2.9.4.2.2.2 Data Fields

Table 105: -

Language Subset Members Table			
Data Field	**Field Type**	**Permitted characters**	**Length**
LinkedId	*SCTID*	Null	0

This field is not used and may be treated as null.

3.2.9.4.2.3 Realm Concept Subset Members Table

3.2.9.4.2.3.1 Key Fields

The *Realm Concept Subset* Members Table has the following key fields:

- *SubsetId*
- *MemberId*
- *MemberStatus*

3.2.9.4.2.3.1.1 SubsetId

Table 106: -

***Realm Concept Subset* Members Table**			
Key Fields	**Field Type**	**Permitted characters**	**Length**
SubsetId	*SCTID*	0 to 9	6 to 18

The unique *SNOMED CT Identifier* for the *Subset* to which this applies.

3.2.9.4.2.3.1.2 MemberId

Table 107: -

***Realm Concept Subset* Members Table**			
Key Fields	**Field Type**	**Permitted characters**	**Length**
MemberId	*SCTID*	0 to 9	6 to 18

The *Identifier* of a *SNOMED CT Concept*.

3.2.9.4.2.3.1.3 MemberStatus

Table 108: -

***Realm Concept Subset* Members Table**			
Key Fields	**Field Type**	**Permitted characters**	**Length**
MemberStatus	*Integer*	0 to 9	1 to 5

The *status* of the identified member in this *Subset*.

The value of *MemberStatus* must be greater than zero.

The priority assigned to this *Concept*.

Note:

The lower the value, the greater the priority. Thus the highest priority is assigned to *Concepts* with the *MemberStatus* value "1" (first).

The priority should be used by applications to determine the items to be displayed first or selected most readily.

3.2.9.4.2.3.2 Data Fields

Table 109: -

***Realm Concept Subset* Members Table**			
Data Fields	**Field Type**	**Permitted characters**	**Length**
LinkedId	*SCTID*	Null	0

This field is not used and may be treated as null.

3.2.9.4.2.4 Realm Description Subset Members Table

3.2.9.4.2.4.1 Key Fields

The *Realm Description Subset* Members Table has the following key fields:

- *SubsetId*
- *MemberId*
- *MemberStatus*

3.2.9.4.2.4.1.1 SubsetId

Table 110: -

***Realm Description Subset* Members Table**			
Key Fields	**Field Type**	**Permitted characters**	**Length**
SubsetId	*SCTID*	0 to 9	6 to 18

The unique *SNOMED CT Identifier* for the *Subset* to which this applies.

3.2.9.4.2.4.1.2 MemberId

Table 111: -

***Realm Description Subset* Members Table**			
Key Fields	**Field Type**	**Permitted characters**	**Length**
MemberId	*SCTID*	0 to 9	6 to 18

The *Identifier* of a *SNOMED CT Description*.

3.2.9.4.2.4.1.3 MemberStatus

Table 112: -

***Realm Description Subset* Members Table**			
Key Fields	**Field Type**	**Permitted characters**	**Length**
MemberStatus	*Integer*	0 to 9	1 to 5

The *status* of the identified member in this *Subset*.

The value of *MemberStatus* must be greater than zero.

Realm Subset The *MemberStatus* specifies the role of the referenced *Description* in this *Realm*. There are three roles.

1 = *Preferred Term*
2 = *Synonym*
3 = *Fully Specified Name*

Note: The following rules must apply to each *Subset*:

1. A *Description* cannot be assigned the role *Fully Specified Name* unless its *DescriptionType* is also "*Fully Specified Name* ";
2. Only one *Description* of each *Concept* may be assigned as the *Fully Specified Name*;
3. If no *Description* of a *Concept* is assigned the role *Fully Specified Name*, the *Fully Specified Name* in the *Concepts Table* assumes this role;
4. A *Description* cannot be assigned the role *Preferred Term* or *Synonym* if its *DescriptionType* is "*Fully Specified Name* ";
5. One and only one *Description* of each *Concept* must be assigned the role "*Preferred Term* ";
6. Any number of *Descriptions* of a *Concept* may be assigned the role "*Synonym* ".

3.2.9.4.2.4.2 Data Fields

Table 113: -

***Realm Description Subset* Members Table**			
Data Fields	**Field Type**	**Permitted characters**	**Length**
LinkedId	*SCTID*	Null	0

This field is not used and may be treated as null.

3.2.9.4.2.5 Realm Relationship Subset Members Table

3.2.9.4.2.5.1 Key Fields

The *Realm Relationship Subset* Members Table has the following key fields:

- *SubsetId*
- *MemberId*
- *MemberStatus*

3.2.9.4.2.5.1.1 SubsetId

Table 114: -

***Realm Relationship Subset* Members Table**			
Key Fields	**Field Type**	**Permitted characters**	**Length**
SubsetId	*SCTID*	0 to 9	6 to 18

The unique *SNOMED CT Identifier* for the *Subset* to which this applies.

3.2.9.4.2.5.1.2 MemberId

Table 115: -

***Realm Relationship Subset* Members Table**			
Key Fields	**Field Type**	**Permitted characters**	**Length**
MemberId	*SCTID*	0 to 9	6 to 18

The *Identifier* of a *SNOMED CT Relationship*.

3.2.9.4.2.5.1.3 MemberStatus

Table 116: -

***Realm Relationship Subset* Members Table**			
Key Fields	**Field Type**	**Permitted characters**	**Length**
MemberStatus	*Integer*	0 to 9	1 to 5

The *status* of the identified member in this *Subset*.

The value of *MemberStatus* must be greater than zero.

Realm Relationship Subset The only permitted value of *MemberStatus* is:

1 = Include

All members of these *Subsets* are treated as equal.

3.2.9.4.2.5.2 Data Fields

Table 117: -

***Realm Relationship Subset* Members Table**			
Data Fields	**Field Type**	**Permitted characters**	**Length**
LinkedId	*SCTID*	Null	0

This field is not used and may be treated as null (for future use).

3.2.9.4.2.6 Context Concept Subset Members Table

3.2.9.4.2.6.1 Key Fields

The *Context Concept Subset* Members Table has the following key fields:

- *SubsetId*
- *MemberId*
- *MemberStatus*

3.2.9.4.2.6.1.1 SubsetId

Table 118: -

***Context Concept Subset* Members Table**			
Key Fields	**Field Type**	**Permitted characters**	**Length**
SubsetId	*SCTID*	0 to 9	6 to 18

The unique *SNOMED CT Identifier* for the *Subset* to which this applies.

3.2.9.4.2.6.1.2 MemberId

Table 119: -

***Context Concept Subset* Members Table**			
Key Fields	**Field Type**	**Permitted characters**	**Length**
MemberId	*SCTID*	0 to 9	6 to 18

The *Identifier* of a *SNOMED CT Concept*.

3.2.9.4.2.6.1.3 MemberStatus

Table 120: -

***Context Concept Subset* Members Table**			
Key Fields	**Field Type**	**Permitted characters**	**Length**
MemberStatus	*Integer*	0 to 9	1 to 5

The *status* of the identified member in this *Subset*.

The value of *MemberStatus* must be greater than zero.

The priority assigned to this *Concept*.

Note:

The lower the value, the greater the priority. Thus the highest priority is assigned to *Concepts* with the *MemberStatus* value "1" (first).

The priority should be used by applications to determine the items to be displayed first or selected most readily.

3.2.9.4.2.6.2 Data Fields

Table 121: -

***Context Concept Subset* Members Table**			
Data Fields	**Field Type**	**Permitted characters**	**Length**
LinkedId	*SCTID*	Null	0

This field is not used and may be treated as null.

3.2.9.4.2.7 Context Description Subset Members Table

3.2.9.4.2.7.1 Key Fields

The *Context Description Subset* Members Table has the following key fields:

- *SubsetId*
- *MemberId*

- *MemberStatus*

3.2.9.4.2.7.1.1 SubsetId

Table 122: -

***Context Description Subset* Members Table**			
Key Fields	**Field Type**	**Permitted characters**	**Length**
SubsetId	*SCTID*	0 to 9	6 to 18

The unique *SNOMED CT Identifier* for the *Subset* to which this applies.

3.2.9.4.2.7.1.2 MemberId

Table 123: -

***Context Description Subset* Members Table**			
Key Fields	**Field Type**	**Permitted characters**	**Length**
MemberId	*SCTID*	0 to 9	6 to 18

The *Identifier* of a *SNOMED CT Description*.

3.2.9.4.2.7.1.3 MemberStatus

Table 124: -

***Context Description Subset* Members Table**			
Key Fields	**Field Type**	**Permitted characters**	**Length**
MemberStatus	*Integer*	0 to 9	1 to 5

The *status* of the identified member in this *Subset*.

The value of *MemberStatus* must be greater than zero.

Context Description Subset The only permitted value of *MemberStatus* is:

1 = Include

All members of these *Subsets* are treated as equal.

3.2.9.4.2.7.2 Data Fields

Table 125: -

Context Description Subset **Members Table**			
Data Fields	**Field Type**	**Permitted characters**	**Length**
LinkedId	*SCTID*	Null	0

This field is not used and may be treated as null.

3.2.9.4.2.8 Navigation Subset Members Table

3.2.9.4.2.8.1 Key Fields

The *Navigation Subset Members Table* has the following key fields:

- *SubsetId*
- *MemberId*
- *MemberStatus*

3.2.9.4.2.8.1.1 SubsetId

Table 126: -

Navigation Subset Members Table			
Key Fields	**Field Type**	**Permitted characters**	**Length**
SubsetId	*SCTID*	0 to 9	6 to 18

The unique *SNOMED CT Identifier* for the *Subset* to which this applies.

3.2.9.4.2.8.1.2 MemberId

Table 127: -

Navigation Subset Members Table			
Key Fields	**Field Type**	**Permitted characters**	**Length**
MemberId	*SCTID*	0 to 9	6 to 18

The *Identifier* of a *SNOMED CT Concept*.

3.2.9.4.2.8.1.3 MemberStatus

Table 128: -

Navigation Subset Members Table			
Key Fields	**Field Type**	**Permitted characters**	**Length**
MemberStatus	*Integer*	0 to 9	1 to 5

The *status* of the identified member in this *Subset*.

The value of *MemberStatus* must be greater than zero.

Navigation Subset The *MemberStatus* specifies the *order* of the *child Concepts* within the set of *Navigation Links* from the same parent *Concept*. The combination of *SubsetId* and *MemberId* and *MemberStatus* forms the unique key.

3.2.9.4.2.8.2 Data Fields

Table 129: -

Navigation Subset Members Table			
Data Fields	**Field Type**	**Permitted characters**	**Length**
LinkedId	*SCTID*	0 to 9	6 to 18

The *ConceptId* of a *Navigation child* of the *Concept* identified by the *MemberId*.

3.2.9.4.2.9 Duplicate Terms Subset Members Table

3.2.9.4.2.9.1 Key Fields

The *Duplicate Terms Subset* Members Table has the following key fields:

- *SubsetId*
- *MemberId*
- *MemberStatus*

3.2.9.4.2.9.1.1 SubsetId

Table 130: -

***Duplicate Terms Subset* Members Table**			
Key Fields	**Field Type**	**Permitted characters**	**Length**
SubsetId	*SCTID*	0 to 9	6 to 18

The unique *SNOMED CT Identifier* for the *Subset* to which this applies.

3.2.9.4.2.9.1.2 MemberId

Table 131: -

***Duplicate Terms Subset* Members Table**			
Key Fields	**Field Type**	**Permitted characters**	**Length**
MemberId	*SCTID*	0 to 9	6 to 18

The DescriptionIdentifier for a *SNOMED CT Concept* that has a *duplicate term*.

3.2.9.4.2.9.1.3 MemberStatus

Table 132: -

***Duplicate Terms Subset* Members Table**			
Key Fields	**Field Type**	**Permitted characters**	**Length**
MemberStatus	*Integer*	0 to 9	1 to 5

The *status* of the identified member in this *Subset*.

The value of *MemberStatus* must be greater than zero.

Duplicate Terms Subset The priority assigned to this *Concept*.

Note:

The lower the value, the greater the priority. Thus the highest priority is assigned to *Concepts* with the *MemberStatus* value "1" (first).

The priority should be used by applications to determine the items to be displayed first or selected most readily.

3.2.9.4.2.9.2 Data Fields

Table 133: -

***Duplicate Terms Subset* Members Table**			
Data Fields	**Field Type**	**Permitted characters**	**Length**
LinkedId	*SCTID*	0 to 9	6 to 18

The MemberID (and therefore, the *DescriptionId*) of the Member with the same *Term* which has the highest priority.

A group of *duplicate terms* can be identified since they will all point to the same *LinkedId*.

3.2.9.4.3 Subset Definition File

Please contact the *IHTSDO* or your *SNOMED CT National Release* Center for the *current* specifications for this file.

3.2.9.5 Cross Maps

3.2.9.5.1 Cross Map Sets Table

Each row in the *Cross Map Sets Table* identifies a *Target Scheme* to which *SNOMED CT* is mapped and specifies characteristics of the associated *Cross Maps* and *Cross Map Targets*.

3.2.9.5.1.1 Key Fields

Table 134: -

Cross Map Sets Table			
Key Fields	**Field Type**	**Permitted characters**	**Length**
MapSetId	*SCTID*	0 to 9	6 to 18

The unique *SNOMED CT Identifier* for this *Cross Map Set*.

Note:

The format and construction of these *Identifiers* is described in *Component features - Identifiers* on page 20.

3.2.9.5.1.2 Data Fields

The *Cross Map Sets Table* contains the following data fields:

- *MapSetName*
- *MapSetType*
- *MapSetSchemeId*
- *MapSetSchemeName*
- *MapSetSchemeVersion*
- *MapSetRealmId*
- *MapSetSeparator*
- *MapSetRuleType*

3.2.9.5.1.2.1 MapSetName

Table 135: -

Cross Map Sets Table			
Data Field	**Field Type**	**Permitted characters**	**Length**
MapSetName	*String*	Any (except LF, CR and TAB)	1 to 255

A descriptive name given to the *Cross Map Set* by its originator.

3.2.9.5.1.2.2 MapSetType

Table 136: -

Cross Map Sets Table			
Data Field	**Field Type**	**Permitted characters**	**Length**
MapSetType	Enumerated	0 to 9	1 to 2

Indicates the nature of the *Cross Maps* associated with this scheme. CrossMapType is used to indicate the inclusion of one to one, one to many and choices of maps.

Table 137: MapSetType Values

Value		**Meaning**
0	Unspecified.	
1	Single	All maps are unique one-to-one maps. • Each *Concept* has only one associated *Cross Map* • Each *Cross Map Target* contains a single *Target Code*.
2	Multiple	Some maps are one-to-many maps but there are no choices. • Each *Concept* has only one associated *Cross Map* • Some *Cross Map Targets* contains a list of more than one *Target Code*.
3	Choice	Some maps include choices of one-to-one maps but there are no one-to-many maps. • Some *Concepts* have more than one associated *Cross Map* • Each *Cross Map Target* contains a single *Target Code*.
4	Flexible	Some maps include choices and there are some one-to-many maps. • Some *Concepts* have more than one associated *Cross Map*.

Some *Cross Map Targets* contain a list of more than one *Target Code*.

3.2.9.5.1.2.3 MapSetSchemeId

Table 138: -

Cross Map Sets Table			
Data Field	**Field Type**	**Permitted characters**	**Length**
MapSetSchemeId	*String*	Any (except LF, CR and TAB)	1 to 64

A standard *Identifier* for the *Target Scheme*. This may be an International Coding Scheme *Identifier* (ISO7826) or an Object *Identifier* (OID) used as specified by *HL7*.

3.2.9.5.1.2.4 MapSetSchemeName

Table 139: -

Cross Map Sets Table			
Data Field	**Field Type**	**Permitted characters**	**Length**
MapSetSchemeName	*String*	Any (except LF, CR and TAB)	1 to 255

The full name of the *Target Scheme*.

3.2.9.5.1.2.5 MapSetSchemeVersion

Table 140: -

Cross Map Sets Table			
Data Field	**Field Type**	**Permitted characters**	**Length**
MapSetSchemeVersion	*String*	Any (except LF, CR and TAB)	1 to 12

The version number of the *Target Scheme* as published by the issuing organization .

3.2.9.5.1.2.6 MapSetRealmId

Table 141: -

Cross Map Sets Table			
Data Field	**Field Type**	**Permitted characters**	**Length**
MapSetRealmId	*String*	Any (except LF, CR and TAB)	1 to 24

A stringidentifying a *Realm* within which this mapping table is applicable. This is only used in cases where *Realm* specific business rules or guidelines alter the acceptable mappings.

Realm is the same as used in the *Subsets Table*. It includes a four character ISO6523 *Identifier* followed by an optional series of concatenated subdivision codes defined by the registered organization .

Example: The EDI *Identifier* scheme used by the *NHS*. The "0080" is the ISO6523 *Identifier* for the *NHS* Trusts, Health Authorities, GP practices are issued with a ten-digit *Identifier* and are permitted to issue subdivision codes of up to five digits in length. This results in a 19-digit *string*.

Usage: This is only used in cases where *Realm* specific business rules or guidelines alter the acceptable mappings. In all other cases the *RealmId* has the value "0" (zero) and should be ignored.

3.2.9.5.1.2.7 MapSetSeparator

Table 142: -

Cross Map Sets Table			
Data Field	**Field Type**	**Permitted characters**	**Length**
MapSetSeparator	*String*	Any (except LF, CR and TAB)	1

The character used as a separator between the individual codes in the *Target Codes* field of the *Cross Map Targets*.

3.2.9.5.1.2.8 MapSetRuleType

Table 143: -

Cross Map Sets Table			
Data Field	**Field Type**	**Permitted characters**	**Length**
MapSetRuleType	Enumerated	0 to 9	1

An indication of the types of rules used in the *Cross Maps* and *Cross Map Targets*.

This discussion of rules is included although rules are not yet used in the released *SNOMED CT Cross Maps*. However, this feature may be of interest to implementers. See *Cross Mapping Guide* on page 208 for more information.

3.2.9.5.1.3 Cross Maps Table

Each row in the *Cross Maps Table* represents one option for mapping a *Concept* to a *Cross Map Target*. If there are several alternative *Cross Maps* for a *Concept*:

- Each option is represented by a row in the *Cross Maps Table*.
- Each row may include rules for choosing that option.
- The rules for choosing an option may be expressed as:
  - Machine-processable instructions;
  - Textual advice to support manual coding;
  - Both of the above.

3.2.9.5.1.3.1 Key Fields

The *Cross Maps Table* contains the following key fields:

- *MapSetId*
- *MapConceptId*
- *MapOption*

3.2.9.5.1.3.1.1 MapSetId

Table 144: -

Cross Maps Table			
Key Fields	**Field Type**	**Permitted characters**	**Length**
MapSetId	*SCTID*	0 to 9	6 to 18

The unique *SNOMED CT Identifier* for the *Cross Map Set* of which this *Cross Map* is a member.

3.2.9.5.1.3.1.2 MapConceptId

Table 145: -

Cross Maps Table			
Key Fields	**Field Type**	**Permitted characters**	**Length**
MapConceptId	*SCTID*	0 to 9	6 to 18

The *SNOMED CT Identifier* of the mapped *Concept*.

3.2.9.5.1.3.1.3 MapOption

Table 146: -

Cross Maps Table			
Key Fields	**Field Type**	**Permitted characters**	**Length**
MapOption	*Integer*	0 to 9	1 to 5

An *integer* that distinguishes between alternative mappings for a single *Concept*.

If automatic rules are used to determine which option is applicable, the *Cross Map* with the lowest *MapOption* value is tested first.

3.2.9.5.1.3.2 Data Fields

The *Cross Maps Table* contains the following data fields:

- *MapPriority*
- *MapTargetId*
- *MapRule*

- *MapAdvice*

3.2.9.5.1.3.2.1 MapPriority

Table 147: -

Cross Maps Table			
Data Field	**Field Type**	**Permitted characters**	**Length**
MapPriority	*Integer*	0 to 9	1 to 5

Indication of the suggested *order* in which to present a series of options for mapping a *Concept* for manual assessment. The first of these is the default option for mapping the *Concept*.

3.2.9.5.1.3.2.2 MapTargetId

Table 148: -

Cross Maps Table			
Data Field	**Field Type**	**Permitted characters**	**Length**
MapTargetId	*SCTID*	0 to 9	6 to 18

The unique *SNOMED CT Identifier* for a *Cross Map Target* to which this *Concept* can be mapped.

3.2.9.5.1.3.2.3 MapRule

Table 149: -

Cross Maps Table			
Data Field	**Field Type**	**Permitted characters**	**Length**
MapRule	*String*	Any (except LF, CR and TAB)	0 to 255

A machine-processable *expression* that determines whether this is an appropriate *Cross Map*.

Note:

The form of *expression* used for these rules depends on the *MapSetRuleType* and is discussed in *Cross Mapping Guide* on page 208.

3.2.9.5.1.3.2.4 MapAdvice

Table 150: -

Cross Maps Table			
Data Field	**Field Type**	**Permitted characters**	**Length**
MapAdvice	*String*	Any (except LF, CR and TAB)	0 to 255

Textual advice to support manual mapping decisions between this *Cross Map* and other alternative *Cross Maps* for mapping the same *Concept*.

3.2.9.5.1.4 Cross Map Targets Table

Each row in this table represents a code or set of codes in the *Target Scheme*, which provides a mapping for one or more *SNOMED CT Concepts*. A *Cross Map Target* may include a rule indicating the conditions in which it applies.

3.2.9.5.1.4.1 Key Fields

Table 151: -

Cross Map Targets Table			
Key Fields	**Field Type**	**Permitted characters**	**Length**
TargetId	*SCTID*	0 to 9	6 to 18

The unique *SNOMED CT Identifier* for the *Cross Map Set* of which this *Cross Map* is a member. See *Component features - Identifiers* on page 20.

3.2.9.5.1.4.2 Data Fields

The *Cross Map Targets Table* contains the following data fields:

- *TargetSchemeId*
- *TargetCodes*
- *TargetRule*
- *TargetAdvice*

3.2.9.5.1.4.2.1 TargetSchemeId

Table 152: -

Cross Map Targets Table			
Data Field	**Field Type**	**Permitted characters**	**Length**
TargetSchemeId	*String*	Any (except LF, CR and TAB)	1 to 64

A standard *Identifier* for the coding scheme in which the TargetCodesare expressed. This may be an International Coding Scheme *Identifier* (ISO7826) or an Object *Identifier* (OID) used as specified by *HL7*.

Note:

The value of this field must be the same as the MapSetSchemeIdof any *Cross Map Set* that includes *Cross Maps* referring to this *Cross Map Target*.

3.2.9.5.1.4.2.2 TargetCodes

Table 153: -

Cross Map Targets Table			
Data Field	**Field Type**	**Permitted characters**	**Length**
TargetCodes	*String*	Any (except LF, CR and TAB)	1 to 255

A code or list of codes in the *Target Scheme* that together represent an appropriate mapping for one or more *Concepts*.

If more than one code is included:

- A separator character specified in the *Cross Map Sets Table* is used to separate the codes;
- The mapping is to the combination of all the codes in the list.

This field may be null if no *target codes* apply to this *Cross Map*.

3.2.9.5.1.4.2.3 TargetRule

Table 154: -

Cross Map Targets Table			
Data Field	**Field Type**	**Permitted characters**	**Length**
TargetRule	*String*	Any (except LF, CR and TAB)	0 to 255

A machine processable *expression* of rules that determine the combinations of conditions to which this *Cross Map Target* applies.

Note:

The form of *expression* used for these rules depends on the Map Set Rule Type and is discussed in *Cross Mapping Guide* on page 208.

3.2.9.5.1.4.2.4 TargetAdvice

Table 155: -

Cross Map Targets Table			
Data Field	**Field Type**	**Permitted characters**	**Length**
TargetAdvice	*String*	Any (except LF, CR and TAB)	0 to 255

Textual advice expressing the combinations of conditions to which this *Cross Map Target* applies.

3.2.9.5.1.5 SNOMED CT - LOINC® Integration Table

Note:

The *LOINC* Integration Table is not currently supported by the *IHTSDO*, and has not been updated for the *current SNOMED CT International Release*. The information below is provided for reference purposes only.

The Laboratory *LOINC* database provides a standardized set of names and codes for identifying laboratory test results. Created by the Regenstrief Institute, the purpose of *LOINC* is to facilitate the exchange and pooling of results for outcomes management and research. Each *LOINC* code represents a unique laboratory test distinguished by six main parts that include the analyte measured, the property observed, the time aspect involved, the sample or system type, the scale of measurement, and where relevant the method of measurement. Taken together, these six components define a single laboratory test result. Additional details about *LOINC*, including access to the *LOINC* database files, are available from *www.regenstrief.org/loinc/loinc.htm*.

SNOMED CT is a clinical *reference terminology*. *SNOMED CT concepts* are explicitly represented in a multi-hierarchical structure. Each of the six distinguishing parts of a given *LOINC* test result are found in *SNOMED CT* and assigned in a *hierarchy*.

The integration exists as a table containing 11 columns. The first nine columns of the table are taken directly from the Laboratory *LOINC* version 1l database. The last two columns contain *concept identifiers* from *SNOMED CT* that relate a specific component of the *LOINC* test to the *SNOMED CT hierarchy*.

An excerpt from the *LOINC* - *SNOMED CT* integration table is shown below. This example shows how *LOINC* code 5792-7 is defined in the integrated *SNOMED CT hierarchy*.

Table 156: Excerpt from SNOMED- LOINC Integration table

LOINC _NUM	**COMPONENT**	**PROPERTY**	**TIME_ ASPCT**	**SYSTEM**	**SCALE_ TYP**	**METHOD _TYP**	**Relationship_ Type**	***ConceptId***
5792-7	GLUCOSE	MCNC	PT	UR	QN	TEST STRIP	116684007	123029007
5792-7	GLUCOSE	MCNC	PT	UR	QN	TEST STRIP	116678009	67079006
5792-7	GLUCOSE	MCNC	PT	UR	QN	TEST STRIP	116680003	69376001

LOINC _NUM	COMPONENT	PROPERTY	TIME_ ASPCT	SYSTEM	SCALE_ TYP	METHOD _TYP	Relationship_ Type	*ConceptId*
5792-7	GLUCOSE	MCNC	PT	UR	QN	TEST STRIP	116686009	122575003
5792-7	GLUCOSE	MCNC	PT	UR	QN	TEST STRIP	116685008	118539007
5792-7	GLUCOSE	MCNC	PT	UR	QN	TEST STRIP	116687000	30766002
5792-7	GLUCOSE	MCNC	PT	UR	QN	TEST STRIP	84203001	117021008

Data in the following columns of the *SNOMED CT - LOINC* Integration table are derived directly from the *LOINC* table:

LOINC_NUM = *LOINC* code

COMPONENT = one of the six parts of the *LOINC fully specified name*; maps to the *SNOMED CT* RelationshipType "has measured component"

PROPERTY = one of the six parts of the *LOINC fully specified name*; maps to the *SNOMED CT* RelationshipType "has property"

TIME_ASPCT = one of the six parts of the *LOINC fully specified name*; maps to the SNOMEDRelationshipType "has time aspect"

SYSTEM = one of the six parts of the *LOINC fully specified name*; maps to the *SNOMED CT* RelationshipType "has specimen"

SCALE_TYP = one of the six parts of the *LOINC fully specified name*; maps to the *SNOMED CT* RelationshipType "has scale type"

METHOD_TYP = one of the six parts of the *LOINC fully specified name*; maps to the *SNOMED CT* RelationshipType "has method"

RELAT_NMS = column in the *LOINC* table that provides *synonyms* (when applicable) for the fully specified *LOINC* name.

ANSWERLIST = column in the *LOINC* table that provides examples of answers for results that are reportable from a multiple choice list (when applicable)

Data in the last two columns of the SNOMED CT - LOINC Integration table are derived directly from the SNOMED CT Concepts Table:

Relationship_Type = *concept identifier* (*ConceptId*) from the *SNOMED CT Concepts* table; defines the *relationship* between the *LOINC* name and the target *SNOMED CT concept*.

ConceptId = target *SNOMED CT Concept* from the *SNOMED CT Concepts* table

The *SNOMED CT - LOINC* integration table may be better visualized with the following figure, where the *SNOMED CT Identifiers* in the last two columns are replaced with their corresponding *SNOMED CT description*.

The table below is an example of *SNOMED CT - LOINC* integration with *SNOMED CT Concept* Ids in last two columns replaced with the *SNOMED CT description*. Underlined cells highlight the part of the *LOINC fully specified name* that is being mapped in each row.

Table 157: SNOMED- LOINC integration example

LOINC_ NUM	COMPONENT	PROPERTY	TIME_ ASPCT	SYSTEM	SCALE_ TYP	METHOD_ TYP	RELAT_ NMS	ANSWER LIST	Relationship_ Type	*ConceptId*
5792-7	GLUCOSE	MCNC	PT	UR	QN	TEST STRIP			Is a	Urinalysis, glucose, qualitative
5792-7	GLUCOSE	MCNC	PT	UR	QN	TEST STRIP			has measured component	Glucose
5792-7	GLUCOSE	MCNC	PT	UR	QN	TEST STRIP			Has property	Mass concentration
5792-7	GLUCOSE	MCNC	PT	UR	QN	TEST STRIP			has time aspect	Single point in *time*
5792-7	GLUCOSE	MCNC	PT	UR	QN	TEST STRIP			Has specimen	Urine specimen
5792-7	GLUCOSE	MCNC	PT	UR	QN	TEST STRIP			Has scale type	Quantitative
5792-7	GLUCOSE	MCNC	PT	UR	QN	TEST STRIP			Has method	Test strip method

The value of the *SNOMED CT* / *LOINC* integration lies within the structure of the corresponding *SNOMED CT* hierarchies.

For example, the first row in the figures above show that *LOINC* 5792-7 is a "Urinalysis, glucose, quantitative" (*ConceptId* 123029007). Looking up *ConceptId* 123029007 ("Urinalysis, glucose, quantitative") in the *SNOMED CT Relationships* Table, the following *hierarchy* can be traced:

- Urinalysis, glucose, quantitative | is a | Urinalysis;
- Urinalysis | is a | Body fluid analysis;
- Body fluid analysis | is a | Laboratory procedure;
- Laboratory Procedure | is a | Procedure.

In the *LOINC* - *SNOMED CT* integration, numerous other Laboratory *LOINC* codes are also defined as | is a | Urinalysis (16408-7, 11135-1, etc.) or, as in the example above, a *child* of Urinalysis (11276-3, 11258-0, etc.). The *SNOMED CT* - *LOINC* integration enables aggregation of *LOINC* -coded data, so that all urinalysis tests, for example, can be isolated. Listed below is a small sample of *LOINC* codes that would be included in a retrieval of all urinalysis procedures.

Table 158: Sample of LOINC codes that would be captured in a retrieval of all Urinalysis procedures

LOINC_NUM	COMPONENT	PROPERTY	TIME_ASPCT	SYSTEM	SCALE_TYP	METHOD_TYP
11276-3	TUBULAR CELLS	ACNC	PT	URNS	ORD	MICROSCOPY.LIGHT
11277-1	SQUAMOUS CELLS	NARIC	PT	URNS	QN	MICROSCOPY.LIGHT.HPF
11278-9	BLADDER CELLS	NARIC	PT	URNS	QN	MICROSCOPY.LIGHT.HPF
11279-7	URINE SEDIMENT COMMENT	FIND	PT	URNS	NAR	MICROSCOPY.LIGHT
12248-1	RENAL CELLS	ACNC	PT	URNS	ORD	MICROSCOPY.LIGHT
12258-0	SQUAMOUS CELLS	ACNC	PT	URNS	ORD	MICROSCOPY.LIGHT.HPF
12448-7	KETONES^ POST CFST	MCNC	PT	UR	QN	TEST STRIP
12512-0	BRUSHITE CRYSTALS	ACNC	PT	URNS	ORD	MICROSCOPY.LIGHT
13653-1	EPITHELIAL CELLS.RENAL	NCNC	PT	URNS	QN	
13945-1	ERYTHROCYTES	NARIC	PT	URNS	QN	MICROSCOPY.LIGHT.HPF
15033-4	ASCORBATE	SRAT	24H	UR	QN	
16408-7	ASCORBATE	MCNC	PT	UR	QN	
1702-0	ACETOACETATE	MCNC	PT	UR	QN	
18407-7	LEUKOCYTES	NRAT	12H	URNS	QN	MICROSCOPY.LIGHT
18487-9	BROAD CASTS	NARIC	PT	URNS	QN	MICROSCOPY.LIGHT.LPF
20409-9	ERYTHROCYTES	NCNC	PT	UR	QN	TEST STRIP
20453-7	EPITHELIAL CELLS	ACNC	PT	URNS	ORD	MICROSCOPY.LIGHT
20409-9	ERYTHROCYTES	NCNC	PT	UR	QN	TEST STRIP

The *SNOMED CT Relationships* Table enables even broader searches such as, for example, all body fluid analyzes . Since Urinalysis is a Body fluid analysis all the *LOINC* codes shown above would be included in the search, in addition to *LOINC* codes for CSF analyzes , joint fluid analyzes , etc.

The *SNOMED CT* - *LOINC* Integration provides a mechanism that reflects the complementary *relationship* between *LOINC* and *SNOMED CT*. This integration yields formally defined appropriately classified laboratory *terms* that can be implemented in the design of robust laboratory analysis applications.

Table 159: Comments on Some LOINC Analyte Names and Their Corresponding SNOMED CT Codes

***LOINC* Analyte Name(s)**	***SNOMED CT* ID** ***SNOMED CT ConceptId***	**Comments**
ENTEROTOXIN	C-00226 116554007	This may represent an assay for bacterial enterotoxins, but rotavirus also produces an enterotoxin, and so we use the more general parent *term* C-00226 Enterotoxin, instead of the more specific *term* C-36205 Bacterial enterotoxin.
ANTHRAQUINONE 9,10-ANTRHACENEDIONE	C-10097 ---------- 116283009	*LOINC* lists 9,10 anthracenedione as a related name for ANTHRAQUINONE, with CAS number 84-65-1 (the number for 9,10anthracenedione. The *SNOMED CT* code means the class of anthraquinones, which includes anthraquinone antineoplastics (e.g. mitoxantrone) as well as anthraquinone laxatives and anthraquinone dyes. We assume an anthraquinone assay would measure any anthraquinone, not just 9,10anthracenedione.
BUTANE N-BUTANE	C-20522 73229004	Butane and n-butane are generally regarded as *synonyms*. In other words, ordinarily isobutane (C-20527) is specified explicitly, and is not counted as a *subtype* of butane.
HEXANE N-HEXANE	C-20524 40647006 C-20529 123008003	Unlike butane, hexane is a general *term* for 6-carbon saturated hydrocarbons. N-hexane is one isoform.
DIOXANE 1,4-DIOXANE	C-21304 27247007	Assume dioxane is 1,4-diethylene dioxide, same as 1,4-dioxane.
HEXACHLOROCYCLOHEXANE GAMMA HEXACHLOROCYCLOHEXANE	C-23112 49490007 C-91170 77496001	*LOINC* lists CAS 608-73-1, but this refers to hexachlorocyclohexane mixed isomers (SNOMEDcode C-20885). We assume a hexachlorocyclohexane assay would measure any isomer or mixture of isomers, so the parent *term* C-23112 is the right one, even though it does not correspond to CAS 608-73-1.

***LOINC* Analyte Name(s)**	***SNOMED CT* ID** ***SNOMED CT ConceptId***	**Comments**
FENCLORVOS FENCHLORPHOS	C-23151 22008009	Fenclorvos is probably a "spelling variant" of fenchlorphos.
ENDOTOXIN	C-36210 18127008	We assume | endotoxin | and | bacterial endotoxin | are synonymous.
KETOCONAZOLE KETOKONAZOLE	C-52B50 40232005	Ketokonazole is probably a "spelling variant".
MONACETYLDAPSONE MONOACETYLDAPSONE	C-55296 123003007	Monacetyldapsone may be a "spelling variant" of monoacetyldapsone.
ACETYLMORPHINE MONOACETYLMORPHINE 6-MONOACETYLMORPHINE	C-606B3 118290009	We assume these are all the same as 6-Omonoacetylmorphine, the major metabolite of heroin (besides morphine). If this is true, it appears to result in some duplicate *LOINC* codes.
OXYCODINONE OXYCODONE	C-606D0 55452001	Oxycodinone is probably another name for oxycodone, but we were unable to find the name Oxycodinone in MEDLINE or any major chemical/drug reference.
6-BETA NALTREXONE 6-BETA NALTREXOL	C-60D22 116562004	The major metabolite of naltrexone is 6-beta naltrexol, also called 6-beta hydroxynaltrexone. We assume this is what is meant by 6-beta naltrexone.
METHABARBITAL METHARBITAL	C-61340 30676006	*LOINC* related names for Methabarbital says Barbitone (not the same! Barbitone is Ethylbarbital!) and also Metharbital (the correct name) is a *synonym*. "Methabarbital" does not occur in MEDLINE, though some lists of controlled substances also use this spelling.
DESMETHYLDOXEPIN NORDOXEPIN	C-62285 96204007	These are *synonyms* for the demethylated metabolite of doxepin, N-desmethyldoxepin.
HYDROXYALPRAZOLAM ALPHA HYDROXYALPRAZOLAM	C-64523 123010001 C-64521 117158005	Probably hydroxyalprazolam is intended to include both alpha- and 4- hydroxyalprazolam... (C-64521 is the alpha form).
N-DESALKYLFLURAZEPAM DESALKYLFLURAZEPAM	C-64552 115558004	*Synonyms* for a major metabolite of flurazepam, N-1-desalkylflurazepam.

***LOINC* Analyte Name(s)**	***SNOMED CT* ID** ***SNOMED CT ConceptId***	**Comments**
DESMETHYLAMIODARONE DESETHYLAMIODARONE	C-80352 96295003	The only important metabolite of amiodarone (particularly in humans) is DESETHYLAMIODARONE, which is the meaning we have chosen here. There are no MEDLINE references containing the word DESMETHYLAMIODARONE, and we have assumed the addition of the "M" may be a spelling error.
LINDANE	C-91170 77496001	Lindane is the gamma isomer of hexachlorocyclohexane (strictly speaking must contain over 99% pure gamma isomer).
AMINOBENZOATE PARA AMINOBENZOATE	C-93210 39707000	Para-aminobenzoic acid (PABA), UV screen, B vitamin. Although both m-aminobenzoic acid and o-aminobenzoic acid exist, they are probably not what is meant here.
FLUBIPROFEN FLURBIPROFEN	C-96050 54344006	Flubiprofen is probably a "spelling variant" of flurbiprofen.
2-AMINOBUTYRATE 2-AMINOBUTYRIC	F-65C04 115343006	2-aminobutyrate.
A-1-IDURONIDASE	F-6A928 86430005	Alpha L iduronidase.
KETOGENIC STEROIDS 17-KETOGENIC STEROIDS	F-B2430 46120009	Assume these mean the same. These steroids have an -OH on C17. They include all of the 17-hydroxycorticosteroids, plus pregnanetriol, cortol, cortolone, etc.
17-HYDROXYKETOSTEROIDS	F-B2430 46120009	Possibly means any steroids with a keto or hydroxy on carbon 17. Assume these are the 17-ketogenic steroids.
20-HYDROXYPROGESTERONE	F-B2460 52422003	F-B2460 is 20-Hydroxyprogesterone, but *LOINC* related names and the CAS number both point to 17-alpha hydroxyprogesterone, which is F-B2470 (and has a keto group at C20!).
11-DEOXYCORTISOL DEOXYCORTISOL	F-B2480 22941009	These two are assumed to be the same.

LOINC Analyte Name(s)	*SNOMED CT* ID *SNOMED CT ConceptId*	Comments
HYDROCORTICOSTERONE 18-HYDROXYCORTICOSTERONE	F-B2491 103034000	Corticosterone is 11,21-Dihydroxypregn-4ene-3,20-dione; hydroxylation of corticosterone could commonly occur at carbons 17 and 18. If a hydroxy is added at 17 we get cortisol, and at 18 we get 18hydrocorticosterone. We assume "hydrocorticosterone" means the latter.
11-OXOPREGNANETRIOL	F-B2746 116617008	Assume this is a class that includes 11ketopregnanetriol plus 11-hydroxy substituted pregnanetriol.
ANDROSTANEDIOL	F-B28A0 103048000	Same as 3-alpha-androstanediol, a major metabolite of dihydrotestosterone. Distinct from the 3-beta-isomer.
HYDROXYCALCIDIOL	F-BB140 1459001	Assume this is 24,25dihydroxycholecaliciferol. Calcidiol is 25hydroxycholecalciferol. The other possibility might be calcitriol, which is 1,25dihydroxycholecalciferol.
COMPLEMENT C3.NEPHRITIC	F-C2402 123001009	*Complement* C3 nephritic factor, is not a *complement* component but an (auto)antibody which causes unregulated C3 splitting activity, usually by preventing intrinsic decay of C3bBb (F-C7640).
COMPLEMENT C4.NEPHRITIC	F-C2403 123002002	*Complement* C4 nephritic factor, is not a *complement* component but an (auto)antibody which prevents the intrinsic decay of C4b2a (F-C7540).
COMPLEMENT CLR2-CLS2 *COMPLEMENT* C1R2+C1S2	F-C70D1 123000005	Assume both refer to C1r2C1s2, see Biochem J 1989 Oct 15;263(2):463-9 "The quaternary structure in solution of human *complement* subcomponent C1r2C1s2".
COMPLEMENT C'2 ESTERASE *COMPLEMENT* C'3 ESTERASE *COMPLEMENT* C'4 ESTERASE	F-C7100 923009 F-C7150 10473000 F-C7200 65044008	Assume these mean the *complement* components C2, C3 and C4. The "prime" symbol (C') used in early literature has been dropped from *complement* nomenclature. C2, C3 and C4 are proenzymes.
COMPLEMENT C3B.INACTIVE	F-C7171 122999009	Could also have *LOINC* name "*COMPLEMENT* IC3B".

LOINC Analyte Name(s)	*SNOMED CT* ID *SNOMED CT ConceptId*	Comments
COMPLEMENT IC3	F-C7172 123004001	Could also have *LOINC* name "*COMPLEMENT* C3.INACTIVE".
COMPLEMENT MEMBRANE C3BC4B COFACTOR PROTEIN	F-C7A10 73042005	Also known as MCP and CD46, a "widely distributed C3b/C4b-binding glycoprotein that inhibits *complement* activation".
HYDROXYTRYPTAMINE SEROTONIN	F-CB040 33635003	Serotonin is 5-hydroxytryptamine. (5hydroxytryptophan is a precursor).
L LITTLE E NOS AG	F-D1300 2728001	There is no Le NOS antigen, but perhaps this means Lewis blood group antigen, not otherwise specified.
ENTEROVIRUS	L-30200 32697003	Picornavirus group.

3.2.9.5.1.5.1 File Structure

SNOMED CT - *LOINC* Integration version 1.0 is provided as an ASCII tab delimited flat file.

Note:

There have been no additions or significant updates to this file since the release of *SNOMED RT* Version 1.1. Minor revisions have been made where *SNOMED CT concepts* have been *retired* and, in some cases, replaced with other *concepts*.

The first row of the file contains column headings.

Table 160: -

Column Name	Data Type	Max. Length	*Description*
LOINC_NUM	Char	7	The unique *LOINC* Code. [10]

[10] Copyright 1995-2008, Regenstrief Institute and the Logical Observation Identifier Names and Codes (LOINC ®) Committee. All rights reserved.

Column Name	**Data Type**	**Max. Length**	***Description***
COMPONENT	Char	150	Fields 2-7 contain the six parts of the name. The fully specified name for a given *LOINC* code would be constructed by printing out the contents of these fields (27), inserting a colon (:) between the contents of each of these fields. [11]
PROPERTY	Char	10	
TIME_ASPCT	Char	10	
SYSTEM	Char	50	
SCALE_TYP	Char	30	
METHOD_TYP	Char	50	
RELAT_NMS	Char	254	One or more synonyms, separated by semicolons (;). This field is intended to make it easier to find a given observation by providing other names by which the observation may be known. For a drug level, for example, we include the trade names of that drug under the related names. [12]
ANSWERLIST	Char	2056	The list of answers for results that are reportable from a multiple choice list (e.g., the answers for the term DISPOSITION OF BLOOD PACK are GIVEN;PARTIALLY GIVEN;DISCARDED). This field provides examples, not required answer lists. [13]
RelationshipType	*Integer*	18	*SNOMED CT Concept Identifier* (*ConceptId*) from the *SNOMED CT Concepts* table. Defines the *relationship* between the *LOINC* name and the target *SNOMED CT concept*.
ConceptId	*Integer*	18	Target *SNOMED CT ConceptId* from the *SNOMED CT Concepts* table

3.2.9.6 History

3.2.9.6.1 History - Introduction

All changes to *SNOMED CT components* are tracked and details of this changes are provided in history files.

The information provided includes details of creation of new components and changes to the *status* of components. When a component is made *inactive*, *relationships* or references are provided to indicate the *active* component that replaces or may be equivalent to the inactivated component.

3.2.9.6.2 Active Concepts

3.2.9.6.2.1 Current Concepts

SNOMED CT Concepts that are intended for *active* use are referred to as *Active Concepts*.

Most *Active Concepts* are suitable for general use and have the *ConceptStatus*"*current*."

Other *Active Concepts* are intended for *active* use in particular circumstances. These *Concepts* have the ConceptStatusvalue "*Pending move*." The significance of this *status* value is described in the following sections.

3.2.9.6.2.2 Concepts Pending Movement to other namespaces

Concepts with the *ConceptStatus*" *Pending move*" are released as *Active Concepts* while waiting for a transitional period after which they are expected to be released in another *namespace*.

SNOMED CT supports the inclusion of *Extensions* to the terminology managed by different organizations . *Concepts* released by different organizations have *ConceptIds* allocated within different *namespaces*. In some situations it is appropriate to transfer maintenance of one or more *Concepts* between separately managed *namespaces*.

After a transfer the original *Concept* becomes *inactive* with the *ConceptStatus* "*Moved elsewhere*." However, the process of release and installation of updates from different *extensions* will typically be asynchronous. Thus a *Concept* move may be initiated in the original *namespace* but not yet completed in the target *namespace*. In this situation, the *Concept* is released with the *ConceptStatus*" *Pending move*." In addition to their *active Descriptions* and *Relationships*, these *Concepts* have a | Moved to |*Relationship* indicating the target *namespace* for the future maintenance of the *Concept*.

If the implementer or user has access to the replacement *Concept* in the target *namespace* then a *Concept* with this *status* may be treated as inactivated. However, unless and until such *Concept* is available, a "*Pending move*" *Concept* should be treated as an *Active Concept*.

3.2.9.6.3 Inactive Concepts

Many *Concepts* released in *SNOMED CT* are, for one *reason* or another, not recommended for *active* use. These *Concepts* are released as part of *SNOMED CT* to support legacy data.

The *ConceptStatus* indicates the *reason* for inactivation of a *Concept*. These include the following:

- ***Retired***: The *Concept* has been withdrawn without a specified *reason*.
- **Duplicate**: The *Concept* has been withdrawn from *active* use because it duplicates another *Concept*.
- **Outdated**: The *Concept* has been withdrawn from *active* use because it is no longer recognized as a valid clinical *Concept*.
- **Ambiguous**: The *Concept* has been withdrawn from *active* use because it is inherently ambiguous.
- **Erroneous**: The *Concept* has been withdrawn from *active* use because it contains an error. A corrected, but otherwise similar, *Concept* has been added to replace it.
- ***Moved elsewhere***: The *Concept* is no longer maintained by the organization responsible for the *namespace* of this *ConceptId*. The *Concept* has been moved to another *namespace* indicated by an associated | Moved to |*Relationship*.

All *Inactive Concepts* are *subtypes* of an appropriate *subtype* of the Special *Concept* "*Inactive concept*." They do no participate in any other *subtype Relationships* and have no *defining characteristic*.

3.2.9.6.3.1 Limited Concepts

Until the January 2010 release of *SNOMED CT*, *Concepts* with the *ConceptStatus* "Limited" were considered to be *Active Concepts* on the basis that they existed in classifications or administrative groupings, and might be needed to meet management or epidemiological objectives. However, they were not recommended for general use due to limitations in the consistency of their interpretation.

Some of these *Concepts* represented aggregations of dissimilar *Concepts* or exception categories intended for otherwise unclassifiable information. Their meaning was dependent on context, and therefore subject to change or reinterpretation that would limit interoperability.

Example: Classifications like *ICD10* include terms containing phrases such as “not otherwise specified” or “not otherwise specified”.

- These are only valid in respect of a particular classification. Another classification or another version of the same classification may include a more precisely defined category that narrows the scope of “not otherwise specified”.
- Many *Concepts* that would fall into “not otherwise classified” categories in *ICD10* can be more precisely represented by another *SNOMED CT Concept*. From a *SNOMED CT* perspective the phrase “not otherwise classified” is therefore open to misinterpretation.

These *Concepts* are now considered to be *inactive* in *SNOMED CT* and are not recommended for any *active* uses.

3.2.9.6.4 Component History Table

Each row in the *Component History Table* represents a single instance of a change to a *component* in a particular release of *SNOMED CT*.

3.2.9.6.4.1 Key Fields

The *Component History Table* has the following key fields:

- *ComponentId*
- *ReleaseVersion*
- *ChangeType*

3.2.9.6.4.1.1 ComponentId

Table 161: -

Component History Table			
Key Fields	**Field Type**	**Permitted characters**	**Length**
ComponentId	*SCTID*	0 to 9	6 to 18

The unique *SNOMED CT Identifier* for the changed *component*.

The *component* may be a *Concept*, *Description*, *Relationship*, *Subset* or a *Cross Map Set*.

3.2.9.6.4.1.2 ReleaseVersion

Table 162: -

Component History Table			
Key Fields	**Field Type**	**Permitted characters**	**Length**
ReleaseVersion	*Integer*	0 to 9	8

The version of *SNOMED CT* in which this change was made.

Several changes may be made during internal maintenance between one release and the next but only the net result of these changes is recorded in the distributed *Component History Table*. Therefore, no more than one change is recorded for each *component* in each *Release Version*.

The format for this field is the *ISO* format date of the release, not the date of the individual change to the component. The format is "yyyymmdd" so for a release on 10 December 2002 the value would be "20021210".

Please refer to *Content History* on page 121 for details about the special considerations for the *Release Version* values used for the *SNOMED CT* First *Release* that merged the *SNOMED RT* and *Clinical Terms Version 3* terminologies.

3.2.9.6.4.1.3 ChangeType

Table 163: Change Type

Component History Table			
Key Fields	**Field Type**	**Permitted characters**	**Length**
ChangeType	Enumerated	0 to 9	1

An indication of the nature of the change.

Table 164: Change Type Values

0	Added	The *component* was added to *SNOMED CT* in this *Release Version*.
1	*Status* Change	The *status* of the *component* has changed since the last *Release Version*.
2	Minor Change	A minor change has been made to this *component* since the last *Release Version*. All other changes require a *component* to be inactivated and replaced with a new component with a different *SCTID*.

3.2.9.6.4.2 Data Fields

The *Component History Table* has the following data fields:

- *Status*
- *Reason*

3.2.9.6.4.2.1 Status

Table 165: -

Component History Table			
Data Fields	**Field Type**	**Permitted characters**	**Length**
Status	Enumerated	0 to 9	1 to 2

The *status* of the *component* after the change.

Table 166: Component Status Values

Value	Name	*Description*
0	*Current*	*active component* In *current* use. This may apply to any type of *component*. However, a *Description* only has this *status* if both it and its associated *Concept* are in *current* use.
1	*Retired*	Inactivated without a specified *reason*.
2	Duplicate	Inactivated as it duplicates another *component*
3	Outdated	Inactivated from *current* use because it is no longer in general clinical use.
4	Ambiguous	*Concept* withdrawn as it is inherently ambiguous.
5	Erroneous	Inactivated as it contains errors. A corrected but otherwise similar *component* has been added to replace it.
6	Limited	Applies to an *Inactive Component* that is based on a classification *concept* or an administrative definition. Also applies to all valid *Descriptions* of that *Concept*.
7	Inappropriate	*Description* has been inactivated as the *Term* should not refer to this *concept*.

Value	Name	*Description*
8	*Concept Inactive*	Remains as a valid *Description* of a *Concept* which is no longer *active*.
9	Implied	*Relationship* withdrawn but is implied by other *active Relationships*.
10	*Moved elsewhere*	The *Concept* has been moved to an *Extension*, to a different *Extension*, or to the *International Release*. Use the | Moved To |*Relationship* to located the *Namespace* to which the *Concept* has been moved. These *Concepts* are considered *inactive*.
11	*Pending move*	The *Concept* will be moved to an *Extension*, to a different *Extension*, or to the *International Release*. Use the | Moved To |*Relationship* to locate the *Namespace* to which the *Concept* will be moved when the recipient organization confirms the move. These *Concepts* are considered *active*.

Note:

1. The permitted values differ according to the nature of the *component*.
2. The release tables contain *active* and *Inactive Concepts* and *Descriptions*. The *status* value in the most recent *Status* Change associated with each *Concept* and *Description* will be identical to the value in the *ConceptStatus* or DescriptionStatusfield for that *component*.
3. *Relationships*, *Subsets*, *Cross Map Sets* and *Cross Map Targets* are only included in a release if their *status* is "*current*". Therefore, the only source of information about the changing *status* of these *components* is the *Component History Table* and *References Table* (for Future Use).
4. The released *Subsets Table* contains *current Subsets* and may also contain previous versions of *Subsets* (distinguished by the *Subset Version*).

3.2.9.6.4.2.2 Reason

Table 167: -

Component History Table			
Data Fields	**Field Type**	**Permitted characters**	**Length**
Reason	*String*	Any (except LF, CR and TAB)	0 to 255

An optional textual *description* of the *reasons* for this change.

3.2.9.6.5 References Table

Each row in the *References Table* 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 Reference indicates the nature of the *relationship* between the *components*.

The *References Table* contains References:

- From each *inactive Description* to one or more other *Descriptions* that are *current* in the *Release Version* in which the *Description* was inactivated;
- From each *inactive Subset* for which there is a *current* replacement to the replacement *Subset*;
- From each *inactive Cross Map Set* for which there is a *current* replacement to the replacement *Cross Map Set*;
- From an *inactive Description* to a *Concept* that is *current* in the *Release Version* in which the *Description* was inactivated, and which is correctly described by the *Term* of the *inactive Description*;
- From an *inactive Concept* with *ConceptStatus* = 10 (*Moved Elsewhere*) to one or more other *Concepts* that are *active* in the *Release Version* in which the *Concept*'s *status* is set to "10".

The *References Table* does **not** include References:

- Between *Concepts*:
  - Except for *Concepts* with *ConceptStatus* = 10, as described above. For all other *Concepts*, equivalent functionality is provided by *Relationships* with specialized *Relationship Types*. The rules for their use for tracking replacement *Concepts* are outlined in *Component History Rules* on page 124.
- Between *Relationships*:
  - There are technical obstacles to tracking *Relationships* that replace or imply other *inactive Relationships* and no practical use case has been identified in which this would provide added value.

The requirement for References between *Cross Map Targets* depends on further analysis and definition of this type of *component*.

3.2.9.6.5.1 Key Fields

The *References Table* has the following key fields:

- *ComponentId*
- *ReferenceType*
- *ReferencedId*

3.2.9.6.5.1.1 ComponentId

Table 168: -

References Table			
Key Fields	**Field Type**	**Permitted characters**	**Length**
ComponentId	*SCTID*	0 to 9	6 to 18

The unique *SNOMED CT Identifier* for the *inactive component*.

The *component* may be a *Description*, a *Subset*, a *Cross Map Set*, or a *Concept* with *ConceptStatus* = 10 (*Moved Elsewhere*).

If the *component* is a *Concept*, then *ReferenceType* may only have a value of "1" (Replaced By) or "4" (Alternative).

Note:

The format and construction of these *Identifiers* is described in *Component features - Identifiers* on page 20.

3.2.9.6.5.1.2 ReferenceType

Table 169: Reference Type

References Table			
Key Fields	**Field Type**	**Permitted characters**	**Length**
ReferenceType	Enumerated	0 to 9	1 to 5

Indicates the nature of the *relationship* between the *inactive component* and the *active component*.

Table 170: Reference Type Values

1	Replaced by	Refers to a revised replacement for the *component*.
2	Duplicated by	Refers to an identical duplicate for the *component*.
3	Similar to	Refers to a *Description* that is identical in all respects except for the associated *Term* which, while not identical, is similar
4	Alternative	Refers to one of several alternatives that are similar or equivalent to the *component* (e.g. where a single *component* is replaced by two more narrowly defined *components*).
5	Moved to	Refers to the *Concept* identifying the target *Namespace* to which a *component* has been moved (*Status* value *Moved Elsewhere* or is scheduled to be moved (*Status* value *Pending Move*).
6	Moved from	Refers to an original *component* in another *Namespace* which is the source of this *current component*.

7	Refers to	*Concept* Refers to a *Concept* which is correctly described by the *Term* of an *inactive Description*.

3.2.9.6.5.1.3 ReferencedId

Table 171: -

References Table			
Key Fields	**Field Type**	**Permitted characters**	**Length**
ReferencedId	*SCTID*	0 to 9	6 to 18

The unique *SNOMED Clinical Terms Identifier* for the referenced *component*.

The *component* identified by the *ReferencedId* must be an instance of the same class of *component* as the component identified by the *ComponentId*, with two exceptions.

1. If the *ReferenceType* value is "5" (Moved to) or "6" (Moved from), *ReferencedId* must identify the *namespace Concept* of the *namespace* that the *component* has been moved to or moved from. This approach is used since the organization sourcing the component may not always be able to determine the precise reference that is applicable in the receiving organization (*namespace*). Thus the responsibility for these references lies with the new responsible (receiving) organization .
2. If the *ReferenceType* value is "7" (Refers to *Concept*), then the *ComponentId* must identify a *Description* and ReferencedIdmust identify a *Concept*.

3.2.9.7 Database View of Distribution Files

3.2.9.7.1 Introduction

The tables in this section provide technical details that could be used to develop a database schema for representing all *SNOMED CT distribution files*. The tables in this section provide a limited set of technical information about each of the fields in the *SNOMED CT distribution files*. The tables also make recommendations about additional keys, which may be necessary to support an effective implementation in a relational database without restructuring the distributed data. However, this information is advisory only and is not a specification of requirements or a recommendation. Alternative approaches based on restructuring the distributed data may provide a more efficient and responsive method of implementation.

Refer to the specification documents for details about the use and possible values of each field.

3.2.9.7.2 Tables

Table 172: Concepts Table

Field Names	Type	Size		Partition	Keys	Use of recommended additional indices		
		***Integer* bits**	***String* length**			**CX1**	**CX2**	**Notes**
ConceptId	*SCTID*	64	18	0	C			**CX1:** To access *Concept* by *CTV3ID* (*Read Code*)
ConceptStatus	Enum	8	2	-				
FullySpecifiedName	*String*	-	255	-				
CTV3ID	*String*	-	5	-		1		**CX2:** To access *Concept* by old style *SNOMEDID*.
SNOMEDID	*String*	-	8	-			1	
Is *Primitive*	*Boolean*	1	1	-				

Table 173: Descriptions Table

Field Names	Type	Size		Partition	Keys	Use of recommended additional indices	
		***Integer* bits**	***String* length**			**DX1**	**Notes**
DescriptionId	*SCTID*	64	18	1	D		**DX1:** To access *Descriptions* for a specified *Concept*
DescriptionStatus	Enum	8	2	-			
ConceptId	*SCTID*	64	18	0	C	1	
Term	*String*	-	255	-			
InitialCapitalStatus	*Boolean*	1	1	-			
DescriptionType	Enum	8	1	-			
LanguageCode	*String*	-	8	-			

Table 174: Relationships tables

Field Names	Type	Size		Partition	Keys	Use of recommended additional indices		
		***Integer* bits**	***String* length**			**RX1**	**RX2**	**Notes**
RelationshipId	*SCTID*	64	18	2	R			**RX1:** To access all or selected *Relationships* of a specified source *Concept*.
ConceptId1	*SCTID*	64	18	0	C	1		

Field Names	Type	Size		Partition	Keys	Use of recommended additional indices		
		***Integer* bits**	***String* length**			**RX1**	**RX2**	**Notes**
Relationship Type	*SCTID*	64	18	0	C	4	3	**RX2:** To access all or selected *Relationships* of a specified target *Concept*.
ConceptId2	*SCTID*	64	18	0	C	5	1	
Characteristic Type	Enum	8	1	-		2	2	
Refinability	Enum	8	1	-				
RelationshipGroup	*Integer*	16	5	-		3		

Table 175: Subsets Table

Field Names	Type	Size		Partition	Keys	Use of recommended additional indices	
		***Integer* bits**	***String* length**			**SX1**	**Notes**
SubsetId	*SCTID*	64	18	3	C		**SX1:** To access previous versions of the same *Subset*
SubsetOriginalID	*SCTID*	64	18	3		1	
SubsetVersion	*Integer*	16	5	-		2	
SubsetName	*String*	-	255	-			
SubsetType	Enum	8	2	-			
LanguageCode	*String*	-	8	-			
RealmID	*String*	-	24	-			
ContextID	*String*	-	18	-			

Table 176: SubsetMembers Table

Field Names	Type	Size		Partition	Primary Key	Other Keys	Use of recommended additional indices		
		***Integer* bits**	***String* length**				**MX1**	**MX2**	**Notes**
Subset Members	Table	-	-	-			G	H	**MX1:** To access all members of *Subset* with the same *MemberStatus*.
SubsetId	*SCTID*	64	18	3	1	S	1	1	
MemberID	*SCTID*	64	18	0, 1, 2	2	X			

Field Names	**Type**	**Size**		**Partition**	**Primary Key**	**Other Keys**	**Use of recommended additional indices**		
		***Integer* bits**	***String* length**				**MX1**	**MX2**	**Notes**
MemberStatus	*Integer*	16	5	-	3[14]		2		**MX2:** To access inverse (*child* to parent) *Navigation Links* in a *Navigation Subset*.
LinkedID	*SCTID*	64	18	0		C		2	

Table 177: Component History Table

Field Names	**Type**	**Size**		**Partition**	**Primary Key**	**Other Keys**
		***Integer* bits**	***String* length**			
Component History	Table	-	-	-		
ComponentID	*SCTID*	64	18	0-5	1	X
ReleaseVersion	*Integer*	32	8	-	2	
ChangeType	Enum	8	1	-		
Status	Enum	8	2	-		
Reason	*String*	-	255	-		

14 TheMemberStatusis only required to be part of the primary key to theSubset Members Table in the case ofNavigation Subsets. In all other cases, the combination of SubsetIdand MemberID must be unique.

Table 178: References Table

Field Names	Type	Size		Partition	Primary Key	Other Keys	Use of recommended additional indices	
		***Integer* bits**	***String* length**				**FX1**	**Notes**
ComponentID	*SCTID*	64	18	0-5	1	D		**FX1:** To access all *components* that referenced a specified *active component*.
ReferenceType	Enum	8	2	-	2			
ReferencedID	*SCTID*	64	18	0-5	3	D	1	

Table 179: Cross Map Sets Table

Field Names	Type	Size		Partition	Primary Key	Other Keys	Use of recommended additional indices	
		***Integer* bits**	***String* length**				**MSX1**	**Notes**
MapSetId	*SCTID*	64	18	4	M			**MSX1:** To access the *Cross Map Set* for a particular alternative coding scheme.
MapSetName	*String*	-	255	-				
MapSetType	Enum	8	2	-				
MapSetSchemeID	*String*	-	64	-		OID	1	
MapSetSchemeName	*String*	-	255	-				
MapSetSchemeVersion	*String*	-	12	-				
MapSetRealmID	*String*	-	24	-				
MapSetSeparator	*String*	-	1	-				
MapSetRuleType	Enum	8	2	-				

Table 180: Cross Maps Table

Field Names	Type	Size		Partition	Primary Key	Other Keys	Use of recommended additional indices	
		***Integer* bits**	***String* length**				**MMX1**	**Notes**
MapSetId	*SCTID*	64	18	4	1	M	1	**MMX1:** To access maps in a stated priority order.
MapConceptId	*SCTID*	64	18	0	2	C	2	
MapOption	*Integer*	16	5	-	3			
MapPriority	*Integer*	16	5	-			3	
MapTargetId	*SCTID*	64	18	5		T		
MapRule	*String*	-	255	-				
MapAdvice	*String*	-	255	-				

Table 181: Cross Map Targets Table

Field Names	Type	Size		Partition	Keys
		***Integer* bits**	***String* length**		
TargetId	*SCTID*	64	18	5	T
TargetSchemeID	*String*	-	64	-	
TargetCodes	*String*	-	255	-	
TargetRule	*String*	-	255	-	

Field Names	Type	Size		Partition	Keys
		***Integer* bits**	***String* length**		
TargetAdvice	*String*	-	255	-	

Table 182: Excluded Words Table

Field Names	Type	Size		Partition	Keys
		***Integer* bits**	***String* length**		
LanguageCode	*String*	-	8	-	1
Keyword	*Integer*	-	8	-	2

Table 183: Description Word Keys Table

Field Names	Type	Size		Partition	Primary Key	Other Keys
		***Integer* bits**	***String* length**			
Keyword	*String*	-	8	-	1	
DescriptionId	*SCTID*	64	18	1	2	D

Table 184: Concept Word Keys Tables

Field Names	Type	Size		Partition	Primary Key	Other Keys
		***Integer* bits**	***String* length**			
Keyword	*String*	-	8	-	1	
ConceptId	*SCTID*	64	18	0	2	C

Table 185: Description Dual Keys Table

Field Names	Type	Size		Partition	Primary Key	Other Keys
		***Integer* bits**	***String* length**			
Dualkey	*String*	-	6	-	1	
DescriptionId	*SCTID*	64	18	1	2	D

Table 186: Concept Dual Keys Table

Field Names	Type	Size		Partition	Primary Key	Other Keys
		***Integer* bits**	***String* length**			
Dualkey	*String*	-	6	-	1	
ConceptId	*SCTID*	64	18	0	2	C

Table 187: Word Equivalents Table

Field Names	Type	Size		Partition	Keys	Use of recommended additional indices	
		***Integer* bits**	***String* length**			**EX1**	**Notes**
WordBlockNumber	*Integer*	32	10	-	1		**EX1:**To access all instances of a particular word or phrase to allow the relevant blocks to be identified.
WordText	*String*	-	8	-	2	1	
WordType	Enum	8	2	-		2	
WordRole	Enum	8	2	-			

Table 188: Canonical Table

Field Names	Type	Size		Partition	Primary Keys	Other Keys	Use of recommended additional indices		
		***Integer* bits**	***String* length**				**CRX1**	**CRX2**	**Notes**
ConceptId1	*SCTID*	64	18	0	1		1		**CRX1:** To access all or selected canonical *Relationships* of a specified source *Concept*.
RelationshipType	*SCTID*	64	18	0	2		3	2	
ConceptId2	*SCTID*	64	18	0	3			1	**CRX2:** To access all or selected canonical *Relationships* to a specified target *Concept*.
RelationshipGroup	*Integer*	16	5	-	4		2		

3.2.9.7.3 Key to the Tables

3.2.9.7.3.1 Type

One of the five possible data types used in *SNOMED CT distribution files*:

- *SCTID* (*SNOMED Clinical Terms Identifier*).
- *String* - A text *string*:
    - Represented as a sequence of *UTF-8* encoded characters.
- *Integer* - An *integer*:
    - Represented as a sequence of digits encoded as UTF-characters.
- Enum - An *integer* which refers to a list of enumerated options:
    - Represented as a sequence of digits encoded as UTF-characters.
- *Boolean* - A *Boolean* value (true or false):
    - Represented as "0" (false) or "1" (true) encoded as a *UTF-8* character.

3.2.9.7.3.2 Size

The size of the data represented as:

- The number of bits in an *integer* representation:
    - This is not applicable to the *string* data type;
    - Possible values 1, 8, 16, 32 or 64.
- The maximum number of characters required for a *string* representation:
    - This is the maximum character length in the distribution file.

3.2.9.7.3.3 Partition

This is only applicable to the *SCTID* data type. It specifies the partition (or in some cases set or range of possible partitions) of the *SCTIDs* used in that field.

3.2.9.7.3.4 Keys

Colored blocks in the keys column identify keys that may be required or useful for effective implementation:

- Primary keys (blue).
- Recommended keys:
    - Critical fields (red) are required for the relevant functionality;
    - User fields (green) are not essential but may improve functionality.
- Foreign keys used in joins to other tables (purple).

A number is used to indicate the *order* of fields in combined keys. A letter in the primary key identifies this as a target for a foreign key pointer. The same letter in a foreign key indicates a join based on the primary key identified by the same letter. Foreign keys marked "X" may refer to any component with an *SCTID* primary key.

3.2.10 The Stated relationships tables

The Stated *Relationships Table* contains the *stated form* of *SNOMED CT*. The *stated form* of a *Concept* is the logic definition that is directly edited by authors or editors. It consists of the stated |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 **|** Is a **|** *relationships* and zero or more defining *relationships*.

The Stated *Relationships Table* is in the same table format as the *Relationships Table*, but the value of the *characteristicTypeId* field is **|** Stated relationship (core metadata concept) **|**.

The *stated form* enables implementers to test a *classifier* for consistency, by comparing the results of classification with the distributed *Relationships Table*, which is the inferred form.

3.2.10.1 Stated relationships tables Summary

Each row in the Stated *Relationships Table* represents a *Relationship* between two *Concepts*.
The Stated *Relationships Table* differs from the main *Relationships Table* in that it only contains those *Relationships* that are directly asserted by authors or editors.

The Stated *Relationships Table* in RF2 has no structural differences from that of the main *Relationships Table*, but the value of the *characteristicTypeId* field is **|** Stated relationship (core metadata concept) **|**.

3.2.10.2 Transforming into OWL or KRSS

Stated *relationships* can be transformed into a *Description Logic* form that utilizes standard syntaxes such as OWL or KRSS. A Perl script to perform the *transformation* from the Stated *Relationships Table* into KRSS and OWL formats is provided as a separate file in the distribution.

3.2.11 Cross Mapping Guide

SNOMED Clinical Terms ® (*SNOMED CT* ®) contains Cross Mapping tables (perhaps more correctly referred to as cross-reference tables) from clinical *concepts* in *SNOMED CT* to categories listed in classifications such as ICD-9-CM.

These classifications are used for data aggregation and analysis and are used by health care systems and services around the world. Effective and efficient use of these files requires an understanding of the rules, conventions and principles of the classification as well as *SNOMED CT* itself.

This section describes the principles used in developing these *Cross Mappings*, the technical schema and a number of examples to communicate how the *SNOMED CT Cross Mapping* could be used to translate from *SNOMED CT* to target mapping(s).

These *Cross Mapping* files are available for the following classifications:

Table 189: Released Cross Map Sets

Classification		Version
ICD-9-CM	International Classification of Diseases 9th Revision Clinical Modification	Updated Version for 2010-2011
ICD-O Topography	International Classification of Diseases for Oncology Geneva, World Health Organization	Version 2, Version 3

The Cross Mapping approach in *SNOMED Clinical Terms* is based on these general principles.

- *SNOMED CT concepts* always retain their meaning:
  - *SNOMED CT* is a *concept* -based *reference terminology* where extensive scientific expertise has been utilized to provide a rich work of medical *concepts* along with their *descriptions*, *synonyms*, logical

descriptors and *relationships* to other *concepts*. Therefore, in developing these mappings, *SNOMED CT concepts* have retained their meaning and not the meaning suggested by a classification.

- Automated mapping is a goal:
  - The *Cross Mapping* scheme is designed to support automated *Cross Mapping* (in future releases) by striving for one-to-one mappings when possible and by providing a technical schema that supports rule-based processing.
- Implementation Review of mappings is expected:
  - The mapping files are intended as tools to jumpstart *SNOMED CT* sites in *Cross Mapping* projects. It is expected that each site will review these files and implement *Cross Mapping* functions and values specific to their needs.
  - Some *concepts* may be part of a *SNOMED CT extension* file and can be ignored if that *extension* is not used. These *concepts* are designed with a 10 in the *partition identifier* component of the *concept* ID.

3.2.11.1 Specification of cross-reference schema

The *SNOMED CT* structure for supporting *Cross Maps* includes three tables specific to the *Cross Map* function. The structure of these tables provides tremendous flexibility to support the *expression* and delivery of several types of *Cross Mappings*. The technical structure of these tables is summarized elsewhere in the *Technical Reference Guide*.

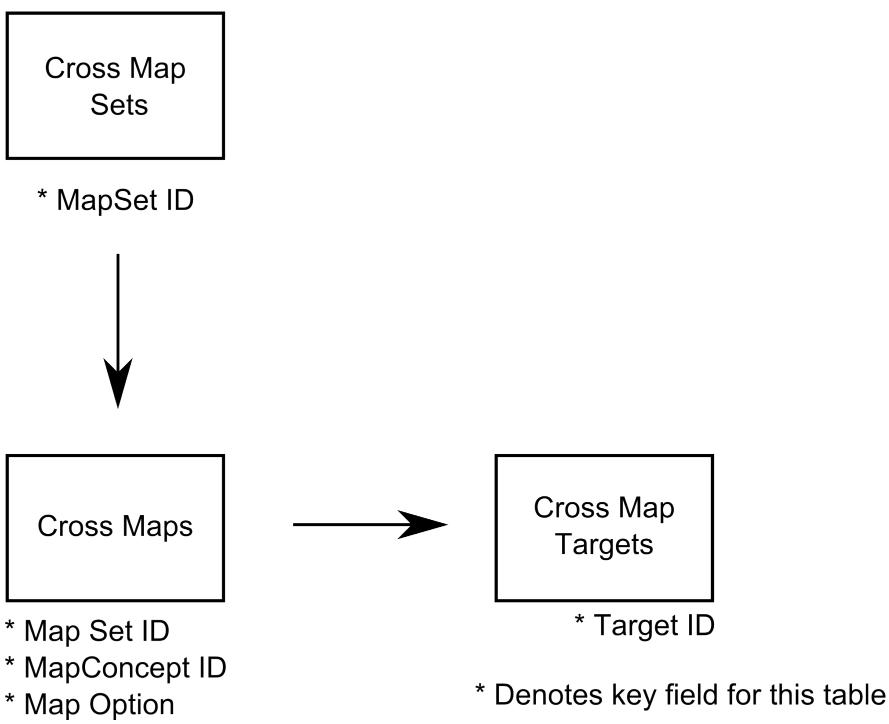


Figure 40: Overview of Cross Map Schema

3.2.11.2 SNOMED CT to ICD-9-CM Epidemiological and Statistical Map

ICD-9-CM, the International Classification of Diseases 9th Revision Clinical Modification Version, is a coding scheme sponsored by the National Center for Health Statistics (NCHS). It is used extensively in the United States for reporting and tracking of mortality, statistical reporting of diseases as well as billing and reimbursement processing. The *SNOMED CT* to ICD-9-CM Cross Map is updated to reflect the version of ICD-9-CM *current* as of the date of its release. It includes all the *SNOMED CT* Clinical Findings (disorders and findings).

Note:

It is important to note that this mapping table is NOT intended for direct billing or reimbursement without additional authoritative review.

Whenever possible, the ICD code or codes(s) with the highest level of specificity have been selected. *Terms* that cannot be assigned to appropriate ICD-9-CM code or codes are considered unmappable. They are denoted by a null in the *Target Codes* field and a zero in the *MapAdvice* field.

3.2.11.2.1 Mapping Categorization Methodology

The *MapAdvice* field in the *Cross Maps* table contains the categorization methodology that provides further information about the characteristics of the map. The methodology is as follows:

Table 190: Mapping Categorization Methodology

Category	*Description*
0	Unmappable. *SNOMED CT term* cannot be assigned to an appropriate ICD-9-CM code.
1	One to one *SNOMED CT* to ICD map. The *SNOMED CT* and ICD correlates are identical or *synonyms*; or the *SNOMED CT term* is listed as an inclusion within the target ICD code *description*. • Many *SNOMED CT concepts* can map to the same ICD *target code description*; • More than one ICD code may be required to fully describe the *SNOMED CT concept*.
2	Narrow to Broad *SNOMED CT* to ICD map. The *SNOMED CT* source code is more specific than the ICD *target code*.
3	Broad to Narrow *SNOMED CT* to ICD map. The *SNOMED CT* code is less specific than the ICD *target code*.
4	Partial overlap between *SNOMED CT* and ICD. Overlap exists between correlates and additional patient information and rules are necessary to select an appropriate mapping.

3.2.11.2.2 SNOMED CT to ICD-9-CM Cross Mapping - Examples

The following examples show the details of how this mapping is delivered.

3.2.11.2.2.1 Sunburn of second degree

This example involves one-to-one mapping from the *SNOMED Clinical Terms concept* Sunburn of Second Degree to the ICD-9-CM category represented by the code "692.76.'

This *concept* is *active* in the *SNOMED CT Concepts* Table.

Table 191: Concepts Table

ConceptId	***Concept Status***	**FullySpecified Name**	***CTV3ID***	***SNOMEDID***	***IsPrimitive***
200834004	0	| sunburn of second degree (disorder) |	M1276	DD-10319	1

The relevant classification is ICD-9-CM.

MapSet Type = 2 means that it is a multiple map in which each unique *SNOMED CT concept* maps to either one *target code* or a set of *target codes*. The *SNOMED CT concept* is never duplicated in more than one row; that is, each *concept* has one and only one map. However, that map can contain more than one *target code*.

Table 192: Cross Maps Sets Table

MapSetId	**MapSet Name**	**MapSet Type**	**MapSet SchemeId**	**MapSet Scheme Name**	**MapSet Scheme Version**	**MapSet *RealmId* ***	**MapSet Separator**	**MapSet Rule Type***
100046	ICD-9-CM Map	2	2.16.840.1.113883.6.5.2.1	International Classification of Diseases and Related Health Problems, 9th Revision, Clinical Modifications.	2006		|	
Fields marked with an asterisk are not currently used in *SNOMED CT*.								

The Map *TargetId* field identifies the row in the Targets Table with the *target codes* for this *SNOMED CT concept*. The *MapAdvice* field contains a value of 1 as the categorization methodology, which, for this *SNOMED CT* to ICD-9-CM mapping, means a 1:1 mapping.

Table 193: Cross Maps Table (Maps Table)

MapSetId **(taken from Maps table above)**	**Map *ConceptId***	**Map Option***	**Map Priority***	**Map *TargetId***	**Map Rule***	**Map Advice**
100046	200834004	0	0	1381051		1
Fields marked with an asterisk are not currently used in *SNOMED CT*.						

The *TargetCodes* are an approximation of the closest ICD-9-CM codes or codes (692.76) that best represent the *SNOMED CT concept*.

Table 194: Cross Map Targets Table (Targets Table)

TargetId	***TargetSchemeId***	***TargetCodes***	***TargetRule ****	***TargetAdvice ****
1381051	2.16.840.1.113883.6.5.2.1	692.76		
Fields marked with an asterisk are not currently used in *SNOMED CT*.				

3.2.11.2.2.2 Pulmonary hypertension with extreme obesity

This *concept* is *active* in the *SNOMED CT Concepts* Table.

Table 195: Concepts Table

ConceptId	***Concept Status***	**FullySpecified Name**	***CTV3ID***	***SNOMEDID***	***IsPrimitive***
276792008	0	| pulmonary hypertension with extreme obesity (disorder) |	Xa0Cx	D3-40324	1

Note:

The MapSet Separator field has a value of bar.

Table 196: Cross Maps Sets Table

MapSetId	MapSet Name	MapSet Type	MapSet SchemeId	MapSet Scheme Name	MapSet Scheme Version	MapSet *RealmId* *	MapSet Separator	MapSet Rule Type*
100046	ICD-9-CM Map	2	2.16.840.1.11 3883.6.5.2.1	ICD-9-CM	2005		|	
Fields marked with an asterisk are not currently used in *SNOMED CT*.								

The *MapAdvice* field contains a value of 2 as the categorization methodology, which means a narrow-to-broad mapping.

Table 197: Cross Maps Table (Maps Table)

MapSetId (taken from Maps table above)	Map *ConceptId*	Map Option*	Map Priority*	Map *TargetId*	Map Rule*	Map Advice
100046	276792008	0	0	329056		2
Fields marked with an asterisk are not currently used in *SNOMED CT*.						

The TargetCodesfield contains approximation of the closest ICD-9-CM codes or codes (416.8, 278.00) that best represent the *concept*. Note that the values are separated by a bar (|), as specified in the MapSet Separator field in the *Cross Maps* Set Table.

Table 198: Cross Map Targets Table (Targets Table)

TargetId	*TargetSchemeId*	*TargetCodes*	*TargetRule* *	*TargetAdvice* *
329056	2.16.840.1.11 3883.6.5.2.1	278.00|416.8		
Fields marked with an asterisk are not currently used in *SNOMED CT*.				

3.2.11.3 SNOMED CT to ICD-O Topography Cross Mapping

The International Classification of Diseases for Oncology (ICD-O) is sponsored by the *World Health Organization* and is used for reporting the topography, morphology and behavior of neoplasms. It is widely used by cancer registries for reporting of cancer cases. The correspondence between ICD-O and *SNOMED CT* consists of two parts: a topography *Cross Map* table, and a numeric morphology correspondence that does not require a separate *Cross Map*.

The *SNOMED CT* to ICD-O Topography map consists of a many-to-one mapping from *concepts* in the *SNOMED CT* Body Structure *hierarchy* to the topography *concepts* in ICD for Oncology (ICDO-2 and ICD-O-3).

Cancer registries receiving *SNOMED CT*-encoded information from health care providers, especially anatomical pathology laboratories, will find this translation helpful. The two most important items of medical information for a cancer patient are the primary site of the tumor and morphologic or histological type of the tumor as diagnosed microscopically by a pathologist.

Topography codes are needed by cancer registries to identify the site of the tumor . The topography *terms* of *SNOMED CT* are more extensive and are not identical to the topography section of ICD-O. *SNOMED CT* includes codes not found in ICD-O; for example, *SNOMED CT* has specific codes for paired anatomic structures for laterality such as left breast while ICD-O does not. In addition, *SNOMED CT* includes *terms* expressed more specifically than in ICD-O such as "subcutaneous and other soft tissues of lower limb and hip" in ICD-O. And finally, *SNOMED CT* includes more *terms* than those needed to identify cancer sites; for example, "nail" is unmappable to ICD-O because the nail is not a site for neoplastic disease.

Reportable cancer cases can also be identified by *SNOMED CT* codes describing tumor morphology and behavior for reportable neoplasms. The ICD-O morphology codes correspond directly with codes that are in the *SNOMEDID* field of the *Concepts Table*. The five-digit ICD-O morphology number (including the final digit behavior code) always begins with 8 or 9, and corresponds exactly to one *SNOMEDID* number that follows the M and the dash. In other words, the row of the *Concepts Table* that corresponds to a given ICD-O morphology code (including the behavior code but without the forward slash) can be identified by an exact match to a *SNOMEDID* that begins with M-8 or M-9. For example, the ICD-O code 8140/3 corresponds to *SNOMEDID* code M-81403. This correspondence is one-to-one for each published ICD-O-3 morphology code, and for this reason, no separate mapping table is necessary for ICD-O morphology.

As a historical note, the morphology correspondence has been maintained in this way continuously since the publication of *SNOMED* Edition 2 in 1979. Further back, the origin of the correspondence is based on the common codes found in SNOP (1965) and MOTNAC, the precursor of ICD-O version 1. However, the SNOP codes had four digits rather than five.

The topography mapping from ICD-O to *SNOMED CT* was a collaborative effort between the College of American Pathologists (CAP), the Centers for Disease Control and Prevention (CDC) and the National Cancer Institute (NCI). The *IHTSDO* would like to acknowledge and offer special thanks to Daniel S. Miller, MD, MPH, former head of the Division of Cancer Prevention and Control, CDC and Mary L. Lerchen, DrPH, MS, Public Health Practice Program Office, CDC who initiated this effort on behalf of cancer registries.

3.2.11.3.1 SNOMED CT to ICD-O Cross Mapping - Example

The following example shows how this mapping is delivered.

3.2.11.3.1.1 Gastric fundus structure

This example involves *SNOMED Clinical Terms concept* Gastric fundus structure to the ICD-O classification represented by the code “C16.1”.

This *concept* is *active* in the *SNOMED CT Concept* Table.

Table 199: ConceptsTable

ConceptId	***ConceptStatus***	**FullySpecified Name**	***CTV3ID***	***SNOMEDID***	**Is *Primitive***
414003	0	| gastric fundus structure (body structure) |	X7551	T-57400	1

The relevant classification is ICD-O.

MapSet Type = **1** means that all maps are one-to-one mappings. The *SNOMED CT concept* is never duplicated in more than one row; that is, each *concept* has one and only one map.

Table 200: Cross Map Sets Table

Map SetId	Map Set Name	Map Set Type	MapSet SchemeId	MapSetS cheme Name	Map Set Scheme Version	Map Set *RealmId* *	Map Set Separator	Map Set Rule Type*
102041	ICD-O-3	1	2.16.840.1.1 13883.6.5.2.2	International Classification of Diseases for Oncology, 3rd Edition	2001	|		
Fields marked with an asterisk are not currently used in *SNOMED CT*.								

Table 201: Cross Maps Table

MapSetId	*MapConceptId*	*MapOption* *	*MapPriority* *	*MapTargetId*	*MapRule* *	*MapAdvice*
102041	414003	0	1	2977055		
Fields marked with an asterisk are not currently used in *SNOMED CT*.						

The *TargetCodes* are an approximation of the closest ICD-O codes or codes (C16.1) that best represent the *SNOMED CT concept*.

Table 202: Cross Map Target Table

TargetId	*TargetSchemeId*	*TargetCodes*	*TargetRule* *	*TargetAdvice* *
2977055	2.16.840.1.11 3883.6.5.2.2	C16.1		
Fields marked with an asterisk are not currently used in *SNOMED CT*.				

3.2.11.4 SNOMED CT - LOINC

Please see *SNOMED CT - LOINC Integration Table* on page 177 for details about *SNOMED CT - LOINC* integration.

3.2.11.5 Cross Mapping Rules

The objective of the *MapRule* fields is to allow addition of machine-processable instructions that then allow automated *Cross Mappings* even in cases where there is more than one way to map a particular *SNOMED CT Concept*. For *Concepts* with several alternative *Cross Maps*, the rules contained in each *Cross Map* should be checked to determine which is most appropriate in the context of a particular patient record.

This section provides an initial syntax of mapping rules and discussion of some possible variants.

Caution: Please contact the *IHTSDO* before using the rules document in this section. Recent work has been done on planning for rules-based maps to ICD-9-CM and *ICD-10*, which may result in a substantial revision of this section.

Note: The discussion of rules is included although rules are not yet used in the released *SNOMED CT Cross Maps*. However, this feature may be of interest to implementers.

3.2.11.5.1 Nature of the Rules

The types of rules required may vary according to the nature of the *Target Scheme*. However, the following general types of rule can be recognized in several different types of mapping:

- Age or sex of the person to whom the *Concept* is applied.
- *Qualifiers* associated with the mapped *Concept*.
- Temporal attributes of the mapped *Concept*.
- Coexisting conditions:
  - Including pathological conditions and other conditions (e.g. pregnancy).
- Related statements:
  - Procedure applied to particular condition.
  - Condition treated by particular procedure.
  - Condition arising as a result of particular procedure.
  - Other statements of causation (e.g. condition caused by accident).

3.2.11.5.2 Representation

Each type of rule can be expressed as a function with parameters which when evaluated returns a true or false result (similar to the *NHS* Casemix grouper tools). Alternatively the rules could be expressed as SQL style queries against an agreed common data model (similar to the *NHS* MIQUEST Health *Query Language*). In either case, the end result is a decision that either accepts or rejects a particular mapping.

Whichever form of representation is used the Extensible Markup *Language* (XML) is suggested as a general-purpose syntax because it support flexible *expressions* and facilitates parsing. Thus the content of the *MapRule* is an XML *string* that contains one or more elements. According to the form of *expression* each element may represent either a function with parameters or *query* predicate.

3.2.11.5.3 Processing order

When determining which of a set of *Cross Map* options is to be used for mapping a particular *Concept* a further consideration is the *order* in which the rules are tested.

The following example is used to illustrate each of the options:

Concept “A” can be mapped to one or more of the five possible *Cross Map* codes (“V”,“W”,“X”,“Y”,“Z”)

- “V” applies if the patient does not have co-existing condition “B” or “C”;
- “W” applies if the patient is under aged 60 and has co-existing condition “B”;
- “X” applies if the patient is aged 60 or over and has co-existing condition “B”;
- “Y” applies if the patient is under aged 60 and has co-existing condition “C”;
- “Z” applies if the patient is aged 60 or over and has co-existing condition “C”.

In addition to the five single *target code* options, there are two possible combinations

- “W,Y”and“X,Z” indicating the presence of co-existing conditions “B”and“C” in either age group.

3.2.11.5.3.1 Order Implementation processing

To allow *order* independent processing of *Cross Maps*, the rules associated with all options for mapping a *Concept* must be complete and mutually exclusive. For every possible occurrence of a *Concept* only one *Cross Map* must have a mapping rule that evaluates as true.

If this approach is followed, there is no opportunity to optimize the *order* of processing and no easy way to provide a catch-all default option to be applied if all other rules fail. Several *Cross Maps* may include the same function or predicate as part of their rules and this may require complex evaluations to be performed more than once.

Table 203: Example of Cross Map Order Implementation Processing

Concept	Option	Rule	*TargetId*	*TargetCodes*
“A”	1	NOT Coexists “B” AND NOT Coexists “C”	101050	V
“A”	2	Age < 60 AND Coexists “B” AND NOT Coexists “C”	102056	W
“A”	3	Age >= 60 AND Coexists “B” AND NOT Coexists “C”	103053	X
“A”	4	Age < 60 AND NOT Coexists “B” AND Coexists “C”	104055	Y
“A”	5	Age >= 60 AND NOT Coexists “B” AND Coexists “C”	105059	Z
“A”	6	Age < 60 AND Coexists “B” AND Coexists “C”	106058	W, Y
“A”	7	Age >= 60 AND Coexists “B” AND Coexists “C”	107054	X, Z

3.2.11.5.3.2 Sequential Processing

The sequential approach tests the rules of each *Cross Map* in a stated *order* until a rule is found to be true. When a true rule is identified the associated *Cross Map Target* is used.

One advantage of this approach is that some optimization can occur to limit the number of times that the same function or predicate is tested. A catch-all default *Cross Map* can be included as the last option in the sequence with a rule which always evaluates as true.

The disadvantage of this approach is that the *order* is significant. Therefore, whenever an alternative *Cross Map* is added the existing *Cross Maps* for that *Concept* must be revised and reordered in a way that achieves the intended results.

Table 204: Example of Sequential Cross Map Rule Processing

Concept	Option	Rule	*TargetId*	*TargetCodes*
“A”	1	Age < 60 AND Coexists “B” AND Coexists “C”	106058	W, Y
“A”	2	Age >= 60 AND Coexists “B” AND Coexists “C”	107054	X, Z
“A”	3	Age < 60 AND Coexists “B”	102056	W
“A”	4	Age >= 60 AND Coexists “B”	103053	X

Concept	Option	Rule	*TargetId*	*TargetCodes*
“A”	5	Age < 60 AND Coexists “C”	104055	Y
“A”	6	Age >= 60 AND Coexists “C”	105059	Z
“A”	7	TRUE	101050	V

3.2.11.5.3.3 Branching Model

The branching approach tests each element of a rule separately and if any test fails it provides a specified reference to the next *Cross Map* to be tested. If all the tests are true the associated *Cross Map Target* is used. If any test fails the specified option is tested next.

If this approach is followed significant optimization can occur so that any test is only performed once for each *Concept* mapped. However, one disadvantage is that addition, removal or revision of any *Cross Map* requires a review of the entire set of *Cross Maps* for that *Concept*. Another disadvantage is that branching logic is less transparent than sequential testing. There are greater risks of logical errors requiring debugging of different test cases.

Table 205: Example of the Cross Map Branching Model

Concept	Option	*MapRule*	*TargetId*	*TargetCodes*
“A”	1	Age < 60 (ELSE 4) AND Coexists “B” (ELSE 3) AND Coexists “C” (ELSE 2)	106058	W, Y
“A”	2	TRUE	102056	W
“A”	3	Coexists “C” (ELSE 7)	104055	Y
“A”	4	Coexists “B” (ELSE 6) AND Coexists “C” (ELSE 5)	107054	X, Z
“A”	5	TRUE	103053	X
“A”	6	Coexists “C” (ELSE 7)	105059	Z
“A”	7	TRUE	101050	V

3.2.11.6 Applying rules to Cross Map Targets

The proposed structures allow rules to be associated with *Cross Map Targets* instead of *Cross Maps*. A *Cross Map Target* may be the target of several *Cross Maps*. If there are no rules in a *Cross Map* any rules in the referenced *Cross Map Target* are tested.

This approach allows a more consistent representation of the rules that relate to generating a particular mapping. However, since each *Cross Map Target* is a stand-alone component that may be referenced by *Cross Maps* associated with several different *Concepts* the rules must be complete and *order* independent.

The decision on whether to apply the rules to the *Cross Maps* or the *Cross Map Targets* depends on the nature of the mapping of *SNOMED CT*. In cases where there is a clear-cut set of primary *Concepts* from which mappings are to be generated, it is preferable to reflect this with rules in the *Cross Maps* for those *Concepts*. In less clearly defined cases, it may be better to express the rules in the *Cross Map Target*.

Example

Extending the illustration in the previous section, *Cross Maps* from *Concept* “B” could reference some of the same *Cross Map Targets* as *Concept* “A”. However, in the case of the map from *Concept* “B” the co-existence of condition “A” must be tested rather than “B”.

Concept “B” could be mapped to:

- “W” if the patient is under aged 60 and has co-existing condition “A” but not “C” ;
- “X” if the patient is aged 60 or over and has co-existing condition “A” but not “C” ;
- "W, Y" if the patient is under aged 60 and has co-existing conditions “A”and“C” ;
- "X, Z" if the patient is aged 60 or over and has co-existing conditions “A”and“C” ;
- An additional *Target Code* “S” if neither of the conditions “A”or“C” is present.

Using the sequential approach the *Cross Maps* for *Concept* “B” would be as follows:

Table 206: Cross Map Rules

Concept	**Option**	**Rule**	***TargetId***	***TargetCodes***
“B”	1	Age < 60 AND Coexists “A” AND Coexists “C”	106058	W, Y
“B”	2	Age >= 60 AND Coexists “A” AND Coexists “C”	107054	X, Z
“B”	3	Age < 60 AND Coexists “A”	102056	W
“B”	4	Age >= 60 AND Coexists “A”	103053	X
“B”	5	TRUE	108057	S

Note:

The first four rows *Cross Maps* refer to the same four *Cross Map Targets* that were used in the mapping from *Concept* “A”. However the rules differ because these are *Cross Maps* from *Concept* “B” rather than from *Concept* “A”.

If rules are to be stated for these *Cross Map Targets*, these must be made complete and *order* independent so that they are not dependent on the *Concept* that initiated the mapping.

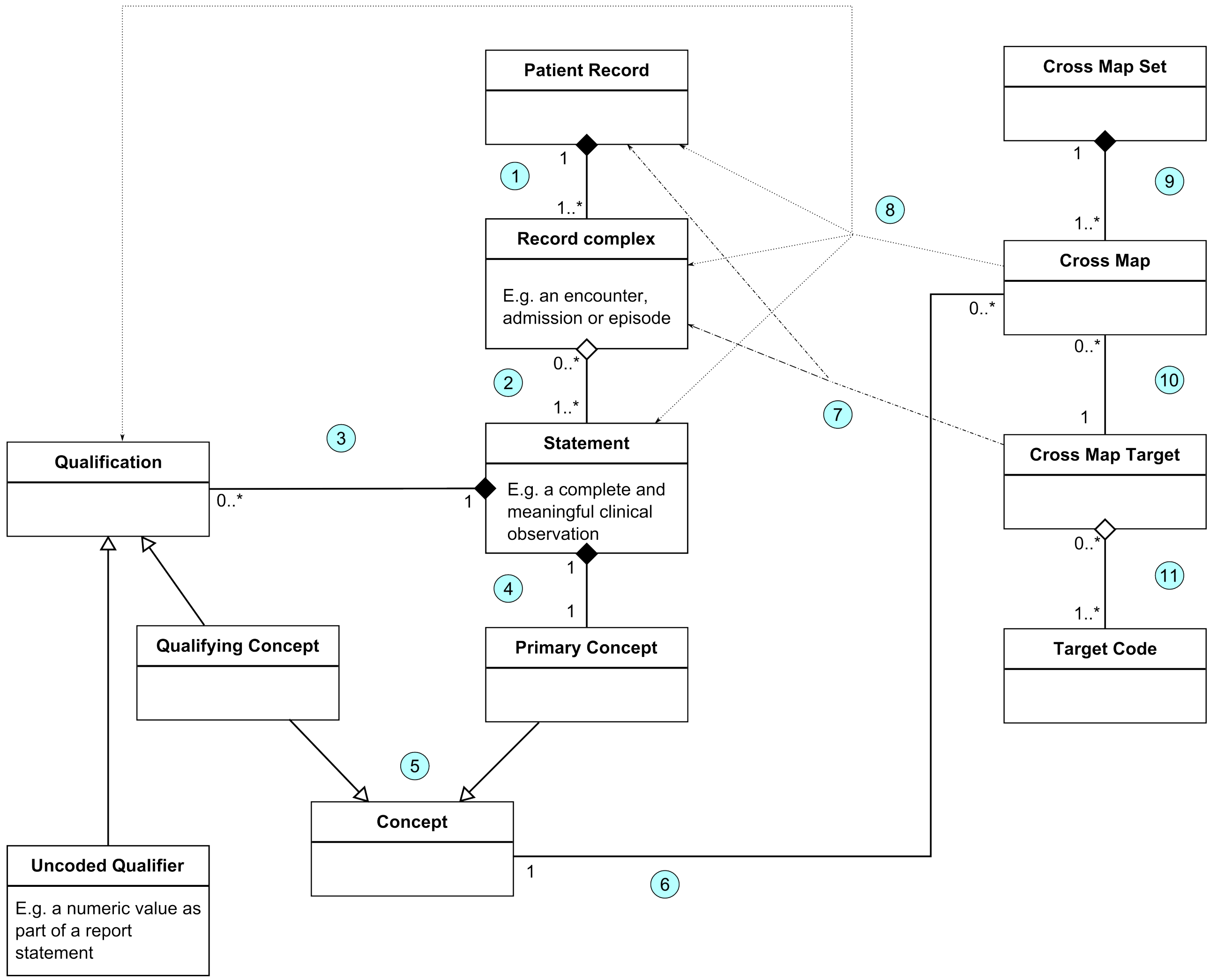
Table 207: Cross Map Target Rules

TargetId	***TargetCodes***	***TargetRule***
101050	V	Coexists “A” AND NOT Coexists “B” AND NOT Coexists “C”
102056	W	Age < 60 AND Coexists “A” AND Coexists “B” AND NOT Coexists “C”
103053	X	Age >= 60 AND Coexists “A” AND Coexists “B” AND NOT Coexists “C”

TargetId	*TargetCodes*	*TargetRule*
104055	Y	Age < 60 AND Coexists “A” AND NOT Coexists “B” AND Coexists “C”
105059	Z	Age >= 60 AND Coexists “A” AND NOT Coexists “B” AND Coexists “C”
106058	W, Y	Age < 60 AND Coexists “A” AND Coexists “B” AND Coexists “C”
107054	X, Z	Age >= 60 AND Coexists “A” AND Coexists “B” AND Coexists “C”
108057	S	Coexists “B” AND NOT Coexists “A” AND NOT Coexists “C”

3.2.11.7 Mapping Model

A logical model of the Cross Mapping process is a general model of the mapping process that pulls together the table structure, patient record, and possible rules processing. The classes *Concept*, *Cross Map Set*, *Cross Map*, and *Cross Map Target* are equivalent to the tables of the same name described earlier in this section. The other classes are shown to illustrate the more detailed aspects of mapping.



1. A record may contain many record complexes.
2. A record complex may contain many statements.
3. A statement may contain a coded or uncoded *qualifier*.
4. A statement contains a primary *concept*.
5. Primary and qualifying *concepts* are specialized uses of *Concepts*.
6. A *Concept* may have any number of *Cross Maps* representing either:

   - mappings to different *Target schemes* or;
   - alternative mappings in the same *Target Scheme*.

   Cross Maps associated with different *Target Schemes* are distinguished by membership of a particular *Cross Map Set*. Alternative *Cross Maps* associated with the same *Target Scheme* are distinguished by examination of other information in the Patient Record either manually or using mapping rules.
7. A *Cross Map Target* may contain rules that fully define it in *terms* of various tests to be applied to a patient record or specified record complex.
8. Each *Cross Map* may contain rules which determine its applicability based on other information in the Patient Record. This may include testing the *qualifiers* applied to the mapped *Concept*, examining other statements in the record including those with specific contextual or temporal *relationships* to the mapped *concept*.
9. A *Cross Map Set* contains all the *Cross Maps* for a particular *Target Scheme*.
10. A *Cross Map* refers to one *Cross Map Target*. Any number of *Cross Maps* may refer to the same *Cross Map Target*.

11. A *Cross Map Target* contains a set of one or more codes in the *Target Scheme*.

Figure 41: A logical model of the Cross Mapping process

3.2.11.7.1 Simple one-to-one mapping

One-to-one mapping between a selected *SNOMED CT Concept* and single *Target Code* is possible in some cases.

This is represented by a single *Cross Map*, which is associated with the *Concept* and is a member of the *Cross Map Set* associated with that *Target Scheme*. The *Cross Map* refers to a *Cross Map Target* that contains only one *Target Code*.

3.2.11.7.2 Simple one-to-many mapping

One-to-many mapping between a selected *SNOMED CT Concept* and set of more than one *Target Codes* is required for *Target Schemes* that use combinations of codes to express the same meaning as a single *SNOMED CT Concept*.

This is represented by a single *Cross Map*, which is associated with the *Concept* and is a member of the *Cross Map Set* associated with that *Target Scheme*. The *Cross Map* refers to a *Cross Map Target* that contains two or more *Target Codes*.

3.2.11.7.3 Mapping with options

Mapping with options is required where there are several possible ways of mapping a *Concept* into the *Target Scheme*.

Multiple options may occur in two circumstances:

- The *Target Scheme* provides more detailed representation than is offered by the *SNOMED CT Concept*. This may occur with *SNOMED CT Concepts* that are mappable if qualified in some way:
  - If the qualification required is an associated qualifying *Concept* or an associated numeric value then automated mapping is feasible;
  - If the qualification is provided as free text then manual intervention will be required but may be assisted by providing appropriate advice.
- The *Target Scheme* encodes the *Concept* in a manner that includes other related information about the patient. Examples of this include:
  - Classifications that subdivide disorders by age, sex or other patient characteristics;
  - Groupers that encode combinations of procedures and disorders;
  - Use of patient record context or links between statements to indicate suspected causation or other clinically significant *relationships*.

These requirements are represented by multiple *Cross Maps* associated with a single *Concept* with the *Cross Map Set* associated with the *Target Scheme*.

- The *Concept* for which *Cross Maps* are sought is the primary *Concept* of a selected Statement in the patient record.
- Where multiple maps are found, the rules in these *Cross Maps* are processed to determine which of the maps is appropriate. These rules may refer to:
  - Qualifications of the selected Statement.
  - Other Statements (including relevant qualifications) that are:
    - Explicitly related to the selected Statement (e.g. a treatment specifically undertaken for a particular condition).
    - In the same complex as the selected Statement (e.g. another diagnosis during the same episode or admission).
    - Temporally related to the selected Statement (e.g. indicating that a patient was pregnant at the time that hypertension was diagnosed).

    - Previous occurrences of the same condition or procedure described in the selected Statement.
    - Related only in being part of the same patient record.
- Other information in the patient record:
    - For example age, sex, occupation, etc.

3.2.12 Text Definitions

The release data includes text definitions for several hundred *SNOMED CT concepts*. To see the definitions, refer to the file in OtherResources/TextDefinitions named "sct1_TextDefinitions_en-US_20110731.txt".

These text definitions are officially part of the *SNOMED CT release*, and are provided to help resolve ambiguities or uncertainties about the meaning a *concept* has in *SNOMED CT*. These definitions refine and clarify what is provided by the *Fully Specified Name*, *Preferred Term*, and the logic definitions (the definitions expressed as *description logic*).

Although many more *concepts* might benefit from text definitions, especially *primitive concepts*, the text definitions file contains all the text definitions available to date.

3.2.12.1 Supplementary Text Descriptions Table

For the *current* version of this table, please refer to the tab-delimited text file in the "Documentation" folder of the *SNOMED CT International Release*.

Table 208: Supplementary Text Descriptions Table

CONCEPTID	*SNOMEDID*	*FULLY SPECIFIED NAME*	DEFINITION
52250000	A-81050	X-ray electromagnetic radiation (physical force)	Electromagnetic radiation of wavelength between approximately 001 nm and 10 nm
56242006	A-81070	Light, electromagnetic radiation (physical force)	Electromagnetic radiation in the visible range as well as parts of the ultraviolet and infrared ranges
75184002	A-81072	Visible light, electromagnetic radiation (physical force)	Electromagnetic radiation in the visible range
55080005	A-81112	Electromagnetic radiation from radar device (physical force)	Electromagnetic radiation from a Radio Detection and Ranging device
55566008	A-A1000	Accidental physical contact (event)	Accidental physical contact or exposure with potential or actual harmful effect
80917008	C-00224	Toxin (substance)	Toxic, noxious, or poisonous substance that is produced by a living organism
128489003	C-200A0	Sand (substance)	Fine granular particles of rock or similar material
119413000	C-20554	Mineral spirits (substance)	Hydrocarbon solvents with flash points above 38 degrees C

CONCEPTID	*SNOMEDID*	*FULLY SPECIFIED NAME*	DEFINITION
56703005	C-21012	Amyl alcohol - commercial (substance)	An alcohol obtained from *refinement* of fusel oil; contains mainly isopentyl alcohol and 2-methyl-1-butanol
88427007	C-21612	Methyl acetylene (substance)	Colorless gas with a sweet odor , used as fuel and shipped as compressed gas
62975006	C-21613	Methyl acetylene-propadiene mixture (substance)	A colorless gas with a characteristic foul odor , used as a fuel and shipped as a liquefied compressed gas
31716004	C-A6032	Frozen plasma product, human (product)	Category of human frozen plasma products, regardless of time from donation to freezing
95323007	D0-00044	Scleredema (disorder)	Hard non pitting edema and induration of the skin; a finding associated with Buschke's disease
410016009	D0-005A3	Lipodermatosclerosis (disorder)	A decrease in lower leg circumference due to recurrent ulceration and fat necrosis causing loss of subcutaneous tissue in a patient with venous stasis disease
367522007	D0-01036	Dermatitis infectiosa eczematoides (disorder)	Inflammation of skin adjacent to an infectious site by autoinnoculation; appears as eczematous plaque with or without vesicles
128045006	D0-01302	Cellulitis (disorder)	Inflammation that may involve the skin and or subcutaneous tissues, and or muscle
402567004	D0-1015C	Vesicular eczema of hands and/or feet (disorder)	Self-limited vesicular eruption of palms and soles
58759008	D0-22020	Intertrigo (disorder)	Superficial dermatitis on opposed skin surfaces
95333004	D0-22138	Eosinophilic pustular folliculitis (disorder)	A dermatosis with pruritic sterile papules and pustules that come together to form plaques with papulovesicular borders, and a tendency toward central clearing and hyperpigmentation, with spontaneous exacerbations and remissions. Histologically variable with folliculitis of follicle sheath and perifollicular dermis and spongiosis of follicular epithelium, sometimes with peripheral leukocytosis and or eosinophilia and or eosinophilic abscesses.

CONCEPTID	*SNOMEDID*	*FULLY SPECIFIED NAME*	DEFINITION
7119001	D0-23010	Cutaneous lupus erythematosus (disorder)	Disease of skin in someone with Lupus erythematosis, though not necessarily systemic or subacute
95336007	D0-40051	Localized	Recurrent ulceration and fat necrosis, associated with loss of subcutaneous tissue and a decrease in lower leg circumference
110986000	D0-53824	Acquired digital fibrokeratoma (disorder)	A keratotic cutaneous polyp containing abundant connective tissue
22649008	D0-75240	Photodermatitis (disorder)	Dermatitis caused by exposure to sunlight
21143006	D0-75310	Calcaneal petechiae (disorder)	Traumatic hemorrhage into heel that persists as black dots
80406003	D1-22350	Pathological dislocation of joint (disorder)	Dislocation of joint caused by presence of another disease
23680005	D1-30000	Enthesopathy (disorder)	Disorder occurring at the site of insertion of tendons or ligaments into bones or joint capsules
109361004	D2-01280	Surgical ciliated cyst (disorder)	A cyst composed of maxillary sinus epithelium along a surgical line of entry
1648002	D2-61424	Lymphocytic pseudotumor of lung (disorder)	Tumor -like mass in lungs composed of fibrous tissue or granulation tissue with inflammatory cells
33622007	D3-16200	Round heart disease (disorder)	A spontaneous cardiomyopathy of unknown etiology that affects healthy poultry
44808001	D3-30000	Conduction disorder of the heart (disorder)	Abnormality in rhythm of heartbeat, including rate, regularity, and/or sequence of activation abnormalities
413577001	D3-80064	Arterial thoracic outlet syndrome due to cervical rib (disorder)	Thoracic outlet syndrome, either nerve or vessel compression, due to a cervical rib
95442007	D3-80506	Peripheral cyanosis (disorder)	Disorder characterized by slowing of blood flow to a body region in association with an increase in oxygen extraction from normally saturated arterial blood

CONCEPTID	*SNOMEDID*	*FULLY SPECIFIED NAME*	DEFINITION
128065004	D4-38016	Congenital partial portal-systemic shunt (disorder)	Congenital portal-systemic shunt in which at least some portal blood perfuses the liver
111029001	D4-40131	Acrokerato-elastoidosis (disorder)	A developmental disorder characterized by keratotic papules of skin of hands and soles with disorganization of dermal elastic fibers that does not appear to be due to trauma or sunlight
111030006	D4-40139	Howel-Evans' syndrome (disorder)	A form of diffuse palmoplantar keratoderma that occurs between the ages of 5 and 15 and may be associated with the subsequent development of esophageal cancer
109478007	D4-51098	Kohlschutter's syndrome (disorder)	Ameliogenesis imperfecta, mental retardation, and epileptic seizures
128533009	D4-A0655	Micropapilla (disorder)	Congenital small optic disc with normal visual function
93040009	D4-A0806	Congenital blepharophimosis (disorder)	A decrease in size of opening of the eye, not due to eyelid fusion, but rather lateral displacement of the inner canthi
94684003	D4-A0824	Microblepharia (disorder)	Congenital abnormal vertical shortness of eyelids
75076004	D4-F1137	Amyelencephalus (disorder)	Congenital absence of the spinal cord and brain
109750005	D5-15106	Abfraction (disorder)	Noncarious lesion, where tooth is fatigued, flexed, and deformed by biomechanical loading of the tooth structure, primarily at the cervical region. These are usually wedge-shaped lesions with sharp-line angles, but sometimes are circular invaginations on occlusal surfaces.
109778006	D5-21244	Bednar's aphthae (disorder)	Symmetric excoriation of the hard palate often due to sucking in infants
109788007	D5-21613	Peripheral ossifying fibroma of gingivae (disorder)	A fibroma of the gums with calcification and possibly ossification
15270002	D5-42004	Obturation obstruction of intestine (disorder)	Complete obstruction of the intestine due to the presence in the lumen of blocking material, such as tumor , fecalith, gallstone, or foreign body

CONCEPTID	*SNOMEDID*	*FULLY SPECIFIED NAME*	DEFINITION
31201001	D5-45211	Knight's disease (disorder)	Infection of perianal region of skin following abrasion, which is named for the occurrence in horsemen
109817001	D5-60862	Intersigmoid hernia (disorder)	Hernia of part of the intestinal tract through the intersigmoid recess or fossa
95564001	D5-90416	Pancreatemphraxis (disorder)	Obstruction of the pancreatic duct leading to swelling of the pancreas as a whole
6595006	D6-34730	Calcinosis (disorder)	Structure with calcium deposition
84757009	DA-30000	Epilepsy (disorder)	A disorder characterized by recurrent seizures
84299009	DA-40020	Neuritis (disorder)	Inflammation of a peripheral and/or cranial nerve
78141002	DA-42100	Erb-Duchenne paralysis (disorder)	A disorder of the superior trunk of the brachial plexus or the fifth and sixth cervical spinal nerves or motor roots, resulting in weakness of proximal upper extremity musculature innervated by these nerve roots
76440000	DA-48100	Equine grass sickness (disorder)	Autonomic dysfunction of unknown etiology in horses, with gut paralysis as primary manifestation
12371008	DA-70170	Ophthalmia nodosa (disorder)	A granulomatous, inflammatory disorder of the eye; reaction to vegetable or insect hairs
95692001	DA-71725	Lipidemia retinalis (disorder)	An abnormal milky appearance of arteries and veins of retina, for example due to lipids in blood greater than 5%, diabetes mellitus, or leukemia
95712005	DA-72546	Entropion uveae (disorder)	Eversion of the margin of the pupil
53889007	DA-73540	Nuclear cataract (disorder)	A cataract involving the nucleus of the lens
44248001	DA-78238	Raymond-Cestan syndrome (disorder)	Abducent nerve paralysis with contralateral hemiparesis
95837007	DC-10190	Central cyanosis (disorder)	A form of cyanosis that occurs when there is a decrease in oxygen saturation in the arterial blood, usually with an SaO2 of below 75%
127062003	DC-38001	Erythrocytosis (disorder)	Peripheral blood red cell count above the normal range

CONCEPTID	*SNOMEDID*	*FULLY SPECIFIED NAME*	DEFINITION
64779008	DC-60000	Blood coagulation disorder (disorder)	Disorders involving the elements of blood coagulation, including platelets, coagulation factors and inhibitors, and the fibrinolytic system
86075001	DC-63000	Coagulation factor deficiency syndrome (disorder)	Includes both quantitative and qualitative disorders of procoagulants
128105004	DC-64101	Von Willebrand disorder (disorder)	Includes true von Willebrand disease with mutation at the VWF locus, as well as mimicking disorders with other mutations (pseudo VWD) and acquired von Willebrand syndrome
128115005	DC-64211	Pseudo von Willebrand disease (disorder)	Any inherited disorder mimicking von Willebrand disease but lacking mutation at the VWF locus
61653009	DD-12432	Bennett's fracture (disorder)	Fracture and dislocation of the first metacarpal and the carpal-metacarpal joint
123976001	DE-00004	Post-infectious disorder (disorder)	A disorder that follows infection but is distinct from the infection itself and its usual manifestations
19168005	DE-01200	Nosocomial infectious disease (disorder)	Infection associated with hospitalization , not present or incubating prior to admission, but generally occurring more than 72 hours after admission
127326005	DF-000D3	Non-human disorder (disorder)	Disorders which occur in animals but not in man
127346000	DF-000F9	Neurologic disorder of eye movements (disorder)	Disorders characterized by eye movement abnormalities that are the result of brain, cranial nerve, or neuromuscular junction dysfunction
417163006	DF-00776	Traumatic AND/OR non-traumatic injury (disorder)	Disorder resulting from physical damage to the body
417746004	DF-00777	Traumatic injury (disorder)	Disorder resulting from physical damage to the body
128207002	DF-00833	Giant axonal neuropathy (disorder)	An autosomal recessive condition characterized by progressive degeneration of the central and peripheral nervous system with enlargement of axons

CONCEPTID	*SNOMEDID*	*FULLY SPECIFIED NAME*	DEFINITION
62014003	DF-10010	Adverse reaction to drug (disorder)	All noxious and unintended responses to a medicinal product related to any dose should be considered adverse drug reactions (from US FDA "Guideline for Industry, Clinical Safety Data Management: Definitions and Standards for Expedited Reporting").
30623001	F-16340	Catch (finding)	A sudden pain, usually sharp, occurring during movement, or exacerbated by movement, and prompting cessation of movement
78064003	F-20000	Respiratory function (observable entity)	Any function involved in the exchange of oxygen and carbon dioxide between the atmosphere and the cells of the body
77329001	F-20160	Mouth breathing (finding)	Habitual breathing through the mouth, usually associated with obstruction of nasal passages
3791008	F-20370	Bohr effect, function (observable entity)	Right shift of the hemoglobin oxygen dissociation curve due to lower pH with increased carbon dioxide
11421009	F-23550	Fremitus (finding)	Vibration felt on the chest wall, either by examiner or subjective
119251002	F-24432	Reverse sneezing (finding)	An inhalation reflex stimulated by an irritation of the mucous membrane of the nose
76777009	F-25170	Artificial respiration by electrophrenic stimulation (procedure)	Procedure that applies electrical stimulation to the phrenic nerve to achieve ventilation
24184005	F-31003	Finding of increased blood pressure (finding)	A finding of increased blood pressure; not necessarily hypertensive disorder
12763006	F-31004	Finding of decreased blood pressure (finding)	A finding of decreased blood pressure; not necessarily hypotensive disorder
85595005	F-31730	Abdominal aortic pulse, function (observable entity)	Pulse felt over the abdominal aorta
61086009	F-31760	Pulse irregular (finding)	A pulse with repeated irregularity
74478000	F-32320	Detection of cardiac shunt (finding)	Anomalous flow of blood between different parts of the circulation
67551009	F-35072	Abnormal third heart sound, S>3< (finding)	Any abnormality of the third heart sound

CONCEPTID	***SNOMEDID***	***FULLY SPECIFIED NAME***	**DEFINITION**
86484008	F-35142	Abnormal fourth heart sound, S>4< (finding)	Any abnormality of the fourth heart sound
32615007	F-35736	Austin Flint murmur (finding)	A mid-to-late diastolic murmur heard best at the cardiac apex, heard in cases of aortic insufficiency
19384000	F-35738	Graham Steell murmur (finding)	High-pitched diastolic murmur heard best at left sternal border, associated with pulmonary valve insufficiency
1735007	F-35760	Thrill (finding)	Vibration felt by examiner on the surface of the body
4592006	F-35820	Fourth sound gallop (finding)	A "galloping" sound on cardiac auscultation because of an abnormally audible fourth heart sound
49864004	F-35830	Protodiastolic gallop with abnormally audible third heart sound (finding)	A "galloping" sound on cardiac auscultation because of an abnormally audible third heart sound
42842009	F-35854	Plateau cardiac murmur (finding)	Cardiac murmur with no significant crescendo or decrescendo
40015002	F-54170	Flatus, function (observable entity)	Passage of gas by anus
119248009	F-62023	Hyperalbuminemia (disorder)	Increased serum albumin concentration
119247004	F-62024	Hypoalbuminemia (disorder)	Reduced serum albumin concentration
102704008	F-63007	Short chain fatty acid (substance)	Fatty acid with fewer than 10 carbon atoms
102705009	F-63008	Medium chain fatty acid (substance)	Fatty acid with 10 to 14 carbon atoms
102706005	F-63009	Long chain fatty acid (substance)	Fatty acid with 10 or more carbon atoms
1677001	F-8A080	Haagensen test (procedure)	Breast examination for malignancy in which patient leans forward and breasts are examined for abnormal contour
91454002	F-A7923	Pleocytosis of cerebrospinal fluid (finding)	Presence of greater than normal number of cells in the cerebrospinal fluid

CONCEPTID	***SNOMEDID***	***FULLY SPECIFIED NAME***	**DEFINITION**
17374005	F-A8870	Queckenstedt's test (procedure)	Measurement of CSF pressure following compression of jugular vein
119249001	F-C0710	Agammaglobulinemia (finding)	Absence of the gamma fraction of serum globulin
119250001	F-C0720	Hypogammaglobulinemia (finding)	Decreased concentration of the gamma fraction of serum globulin
106190000	F-C30F9	Allergic state (disorder)	Known to have allergic reactions to particular substance(s)
70730006	F-D0110	Ineffective erythropoiesis (finding)	Increased destruction of erythrocyte precursors
50857004	F-D7300	Tissue factor (substance)	Tissue factor, the high-affinity receptor and cofactor for the plasma serine protease VII/VIIa
20364005	F-F5002	Paracusis (disorder)	Altered sense of hearing, other than simple decreased hearing or deafness
184034005	F-F5074	Auditory area (sound intensity) (observable entity)	*Range* of sound intensity between the minimum audible intensity and the auditory pain threshold
399264008	G-0373	Image mode (observable entity)	Image mode refers to the type of image acquisition (modality). For example, most ultrasound systems use 2D, Color Flow, M Mode and Doppler modes.
399030000	G-0374	Left ventricular systolic area (observable entity)	Area measurement of the left ventricle in systole.
399109006	G-0375	Left ventricular diastolic area (observable entity)	Area measurement of the left ventricle in diastole.
399287000	G-0376	Left ventricular area fractional change (observable entity)	(Diastolic Area - Systolic Area) / Diastolic Area
399063007	G-0377	Left ventricular semi-major axis diastolic dimension (observable entity)	Semi-major axis is the dimension from the widest minor axis to the apex of the left ventricle at end diastole.
399309003	G-0378	Left ventricular truncated semi-major axis diastolic dimension (observable entity)	Truncated semi-major axis is from widest short axis diameter to the mitral annulus plane in the left ventricle at diastole.

CONCEPTID	*SNOMEDID*	*FULLY SPECIFIED NAME*	DEFINITION
399293008	G-0379	Left ventricular epicardial diastolic area, psax pap view (observable entity)	Epicardial area of the left ventricle, cross section [parasternal short axis view] at the level of the papillary muscles in diastole.
399133000	G-037A	Left ventricular peak early diastolic tissue velocity (observable entity)	Myocardial tissue velocity by Pulsed Wave Doppler, typically adjacent to the mitral annulus, measured in early diastole.
399140004	G-037B	Ratio of mitral valve peak velocity to left ventricular peak tissue velocity e-wave (observable entity)	Transmitral velocity measured at the leaflet tips at the onset of diastole divided by the myocardial velocity measured at the same point in the cardiac cycle. Correlates with left ventricular filling pressure.
399007006	G-037C	Left ventricular peak diastolic tissue velocity during atrial systole (observable entity)	Myocardial tissue velocity by Pulsed Wave Doppler, typically adjacent to the mitral annulus, measured at the time of left atrial contraction.
399167005	G-037D	Left ventricular peak systolic tissue velocity (observable entity)	Myocardial tissue velocity by Pulsed Wave Doppler, typically adjacent to the mitral annulus, measured during left ventricular systole.
399051002	G-037E	Left ventricular isovolumic contraction time (observable entity)	The time interval from mitral valve closure to aortic valve opening. Measured as the interval between the mitral valve closing click and the aortic valve opening click on Continuous Wave Doppler.
399266005	G-037F	Left ventricular index of myocardium performance (observable entity)	(MCO-ET(Left Ventricle OT))/ET(Left Ventricle OT), where MCO is MV Closure to Opening time and ET is Ejection Time.
399023006	G-0380	Right ventricular peak systolic pressure (observable entity)	Right ventricular systolic pressure calculated from the peak right ventricular-right atrial systolic gradient (from the peak tricuspid regurgitation velocity using the modified Bernoulli equation) plus estimated right atrial/central venous pressure.
399154007	G-0381	Right ventricular index of myocardial performance (observable entity)	(TCO-ET(RVOT))/ET(RVOT), where TCO is TV Closure to Opening time and ET is Ejection Time.
399058008	G-0382	Ratio of aortic valve acceleration time to aortic valve ejection time (observable entity)	Ratio of Aortic Valve Acceleration Time to Aortic Valve Ejection *Time*

CONCEPTID	*SNOMEDID*	*FULLY SPECIFIED NAME*	DEFINITION
399235004	G-0383	Left atrium systolic volume (observable entity)	Volume of blood contained in the left atrium at end-systole.
399354002	G-0384	Mitral valve E-wave deceleration time (observable entity)	The time interval from the peak of the transmitral Doppler early filling velocity to the *intersection* with the Doppler baseline derived from the slope of the transmitral early filling wave.
399229004	G-0385	Mitral valve A-wave duration (observable entity)	Duration of the transmitral velocity wave during atrial contraction.
399062002	G-0386	Ratio of mitral valve acceleration time to mitral valve deceleration time (observable entity)	Ratio of the Mitral Valve Acceleration Time to the Mitral Valve Deceleration Time.
399104001	G-0387	Mitral valve closure to opening time (observable entity)	The time interval from the closure of the 1st Doppler spectral taken from the mitral valve to the opening of the 2nd Doppler spectral of the mitral valve.
399238002	G-0388	Ratio of pulmonic valve acceleration time to pulmonic valve ejection time (observable entity)	Ratio of Pulmonic Valve Acceleration Time to Pulmonic Valve Ejection *Time*
399282006	G-0389	Tricuspid valve closure to opening time (observable entity)	The time interval from the closure of the 1st Doppler spectral taken from the tricuspid valve to the opening of the 2nd Doppler spectral of the tricuspid valve.
399048009	G-038A	Main pulmonary artery peak velocity (observable entity)	Peak velocity obtained from Pulsed Wave Doppler or continuous wave Doppler, positioned in the main pulmonary artery.
399070007	G-038B	Pulmonary vein A-wave duration (observable entity)	Duration of the retrograde velocity in the pulmonary vein during atrial contraction.
399267001	G-038C	Pulmonary vein S-wave velocity time integral (observable entity)	The integral of the Doppler spectral profile of the systolic component of pulmonary venous flow.
399039004	G-038D	Pulmonary vein D-wave velocity time integral (observable entity)	The integral of the Doppler spectral profile of the diastolic component of pulmonary venous flow.

CONCEPTID	*SNOMEDID*	*FULLY SPECIFIED NAME*	DEFINITION
399367004	G-038E	Cardiovascular orifice area (observable entity)	Area of an orifice as calculated by Orifice area = Peak Instantaneous Flow Rate / Maximal Velocity of the Regurgitant jet at the Jet Orifice
399301000	G-0390	Regurgitant fraction (observable entity)	Ratio of the Regurgitant Volume to the Stroke Volume: Regurgitant Volume / Inflow Volume. The Regurgitant Volume is the retrograde flow. The Inflow Volume is total antegrade volume. which is the sum of the Regurgitant Volume and net antegrade volume.
399093001	G-0391	Medial mitral annulus structure (body structure)	Area of the mitral annulus adjacent to the left ventricular septum and outflow tract.
399086000	G-0392	Lateral mitral annulus structure (body structure)	Area of the mitral annulus adjacent to the left ventricular posterolateral wall.
399345000	G-0393	Adult echocardiography procedure report (record artifact)	Document title of adult echocardiography procedure (evidence) report.
399339008	G-0395	Apical long axis (*qualifier* value)	Imaging plane with the transducer at the cardiac apex, which includes the left ventricle, left atrium, aortic outflow tract and proximal aorta. Usually visualizes a small portion of the right ventricle in the near field.
399139001	G-0396	Parasternal long axis view (*qualifier* value)	Imaging plane with the transducer at the left sternal border oriented along the long axis of the left ventricle, which includes the left ventricle, left atrium, aortic outflow tract and proximal aorta. Usuallyvisualizes a small portion of the right ventricle.
399306005	G-0397	Parasternal short axis view (*qualifier* value)	Imaging plane with the transducer at the left sternal border oriented along the short axis of the left ventricle.
399239005	G-0398	Parasternal short axis view at the aortic valve level (*qualifier* value)	Imaging plane with the transducer at the left sternal border oriented along the short axis of the left ventricle, at the base of the heart, IVC, atrial septum, tricuspid valve, which includes the aortic valve, right and left atria, right ventricular outflow tract and pulmonic valve, pulmonary artery, and in some patients, the LPA and RPA.

CONCEPTID	*SNOMEDID*	*FULLY SPECIFIED NAME*	DEFINITION
399371001	G-0399	Parasternal short axis view at the level of the mitral chords (*qualifier* value)	Imaging plane with the transducer at the left sternal border oriented along the short axis of the left ventricle, which includes the left ventricle at the level of the mitral chords, ventricular septum, and right ventricle. This plane is inferior to the mitral valve.
399036006	G-039A	Parasternal short axis view at the mitral valve level (*qualifier* value)	Imaging plane with the transducer at the left sternal border oriented along the short axis of the left ventricle, which includes the left ventricle at the level of the mitral valve leaflets, ventricular septum, and right ventricle. This plane is inferior to the aortic valve at the base of the heart.
399271003	G-039B	Parasternal short axis view at the papillary muscle level (*qualifier* value)	Imaging plane with the transducer at the sternal border oriented along the short axis of the left ventricle, which includes the left ventricle at the level of the papillary muscles, and right ventricle. This plane is inferior to the cordae.
398998003	G-039C	Right ventricular inflow tract view (*qualifier* value)	View of the Portion of the right ventricle adjacent to the tricuspid valve, the inflow portion of the RV. Common acronym: RVIT.
399195005	G-039D	Right ventricular outflow tract view (*qualifier* value)	View of the portion of the right ventricle adjacent to the pulmonic valve, the outflow portion of the RV. Common Acronym: RVOT.
399310008	G-039E	Subcostal long axis view (*qualifier* value)	Imaging plane with the transducer at the subcostal space (inferior to sternum) oriented along the long axis of the left ventricle, which includes the left ventricle, left atrium, right ventricle and right atrium, septum between left and right ventricles, and septum between the left and right atria.
399200001	G-039F	Subcostal short axis view (*qualifier* value)	Imaging plane with the transducer at the subcostal space oriented along the short axis of the left ventricle, either at the aortic valve level (base of heart) or more inferior plane of section through the left ventricle
399106004	G-03A0	Suprasternal long axis view (*qualifier* value)	Imaging plane with the transducer in the suprasternal notch, oriented along the long axis of the ascending aorta, aortic arch vessels, and proximal descending aorta.

CONCEPTID	*SNOMEDID*	*FULLY SPECIFIED NAME*	DEFINITION
399145009	G-03A1	Suprasternal short axis view (*qualifier* value)	Imaging plane with the transducer in the suprasternal notch, oriented along the short axis of the aortic arch. This plane of sectionvisualizes a cross-section of the aorta, long axis of the RPA, and in the pediatric patient, the left atrium and four pulmonary veins.
399064001	G-03A2	2D mode ultrasound (*qualifier* value)	2D, B-mode, B-scan image type
123038009	G-8000	Specimen (specimen)	Material (structure, substance, device) removed from a source (patient, donor, physical location, product)
399232001	G-A19B	Apical two chamber view (*qualifier* value)	Imaging plane with the transducer at the cardiac apex, which includes the left ventricle and left atrium.
399214001	G-A19C	Apical four chamber view (*qualifier* value)	A coronal imaging plane with the transducer at the cardiac apex which includes the left ventricle, left atrium, right ventricle and right atrium.
260674002	G-C048	Direction of flow (attribute)	The flow of an anatomic orifice. The orifice may be an opening such as a valve or stenosis. The direction may be valve retrograde flow (regurgitation) or antegrade flow.
119246008	J-14126	Imam (occupation)	Muslim prayer leader
388445009	L-000A9	Genus Equus (organism)	Horse, donkey, mule genus
68014009	L-88105	Canis familiaris (organism)	Domestic dog subspecies
125085001	L-8A122	Equus asinus asinus (organism)	Equus subspecies
125086000	L-8A144	Equus caballus gmelini X Equus caballus caballus (organism)	Intersubspecies equine hybrid
78678003	L-8B100	Sus scrofa (organism)	Domestic pig subspecies
388393002	L-8B1FB	Genus Sus (organism)	Swine genus

CONCEPTID	*SNOMEDID*	*FULLY SPECIFIED NAME*	DEFINITION
125093001	L-8B951	Bison bison X Bos taurus indicus X Bos taurus taurus (organism)	Intergenus hybrid of cattle
125094007	L-8B952	Bos taurus indicus X Bos taurus taurus (organism)	Intersubspecies cattle hybrid
125095008	L-8B953	Bos javanicus X Bos taurus indicus (organism)	Interspecies hybrid of cattle
388168008	L-8BA18	Genus Bos (organism)	Cattle genus
125097000	L-8C306	Capra hircus (organism)	Domestic goat
125099002	L-8C336	Ovis aries (organism)	Domestic sheep species
125101009	L-8C338	Merino sheep superbreed (organism)	Merino sheep breed group
125102002	L-9210A	Anas platyrhynchos (organism)	Mallard duck species
396620009	L-921FA	Genus Anas (organism)	Duck genus
15778005	L-92220	Anser (organism)	Goose genus
70881005	L-92222	Anser anser anser (organism)	Greyleg goose subspecies
125104001	L-9222A	Anser anser (organism)	Greyleg goose species
47290002	L-93790	Gallus gallus (organism)	Junglefowl
125105000	L-9379A	Gallus (organism)	Junglefowl genus
125671007	M-01444	Rupture (morphologic abnormality)	Disruption of continuity of tissue, not necessarily due to external forces; may be due to weakness in the tissue or excessive internal pressures
128176002	M-01471	Cutaneous patch (morphologic abnormality)	Skin lesion, greater than 2 cm, flat, colored ; differs from a macule only in size
128177006	M-01472	Cutaneous plaque (morphologic abnormality)	Skin lesion, greater than 2 cm, flat, colored ; differs from a macule only in size

CONCEPTID	*SNOMEDID*	*FULLY SPECIFIED NAME*	DEFINITION
27925004	M-03010	Nodule (morphologic abnormality)	A 1 to 5 cm firm lesion raised above the surface of the surrounding skin; differs from a papule only in size
25694009	M-03130	Papule (morphologic abnormality)	A small, solid lesion, less than 1 cm in diameter, raised above the surface of the surrounding skin and hence palpable
112629002	M-04013	Macule (morphologic abnormality)	A flat lesion, less than 2 cm in diameter, not raised above the surface of the surrounding skin
19130008	M-10000	Traumatic abnormality (morphologic abnormality)	A structure damaged by an external force
384709000	M-10005	Sprain (morphologic abnormality)	Injury to a ligament due to movement of joint beyond normal range
161006	M-11000	Thermal injury (morphologic abnormality)	Injury due to increased heat
48333001	M-11100	Burn injury (morphologic abnormality)	Generic burn injury, including that due to excessive heat, as well as cauterization , friction, electricity, radiation, sunlight, and other causes
105594005	M-11106	Thermal burn (morphologic abnormality)	Burn injury due to excessive heat
127559009	M-31318	Everted margin (morphologic abnormality)	The structure representing the everted margin of a part
189411005	M-35011	Antemortem thrombus (morphologic abnormality)	Antemortem blood clot in the cardiovascular system
64305001	M-36320	Urticaria (morphologic abnormality)	A raised, erythematous papule or cutaneous plaque, usually representing short-lived dermal edema
82515000	M-36750	Vesicle (morphologic abnormality)	A small (less than 1 cm) fluid-filled lesion, raised above the plane of surrounding skin
339008	M-36760	Blister (morphologic abnormality)	A fluid-filled, raised, often translucent lesion, greater than 1 cm in diameter
47002008	M-41601	Pustule (morphologic abnormality)	A vesicle filled with leukocytes

CONCEPTID	*SNOMEDID*	*FULLY SPECIFIED NAME*	DEFINITION
128419007	M-55090	Pathologic mineralization (morphologic abnormality)	Deposition of mineral in normally non-mineralized tissue
18115005	M-55420	Pathologic calcification, calcified structure (morphologic abnormality)	Deposition of calcium in normally non-calcified tissue
122869004	P0-00081	Measurement procedure (procedure)	An observation, by some objective method, of amount, number, quantity, size, level, extent, or magnitude, resulting in an ordinal or quantitative value
118661008	P0-00098	Physician service (procedure)	Service provided by physician
14734007	P0-00100	Administrative procedure (procedure)	Procedure related to the administrative aspects of health care, including admission, discharge, transfer, disposition, referral, business, legal, financial, quality review, peer review, data reporting, notification, and so forth
363687006	P0-0099C	Endoscopic procedure (procedure)	An inspection done with an endoscope
197157006	P0-0099D	Photography of patient (procedure)	An observation that generates a recording made from energy of the light spectrum
169283005	P0-009A0	Medical photography (procedure)	An observation that generates a recording made from energy of the light spectrum
386053000	P0-009B4	Evaluation procedure (procedure)	determination of a value, conclusion, or inference by evaluating evidence
387713003	P0-009C3	Surgical procedure (procedure)	Planned structural alteration of the body, usually requiring disruption of a body surface (usually skin or mucosa)
410614008	P0-00A65	Construction (procedure)	The act of building something
410619003	P0-00A66	Application (procedure)	Introduction of a substance or device to the surface of the body
409063005	P0-00B19	Counseling (procedure)	Psychosocial procedure that involves listening, reflecting, etc. to facilitate recognition of course of action / solution.

CONCEPTID	*SNOMEDID*	*FULLY SPECIFIED NAME*	DEFINITION
409073007	P0-00B27	Education (procedure)	Procedure that is synonymous with those activities such as teaching, demonstration, instruction, explanation, and advice that aim to increase knowledge and skills, change behaviors , assist coping and increase adherence to treatment.
416118004	P0-00B68	Administration (procedure)	Introduction of a substance to the body
119265000	P0-04001	Assisting (procedure)	Helping the body perform a function it normally does on its own
119270007	P0-04003	Management procedure (procedure)	A plan or recommendation for services, based on an evaluation
122545008	P0-04006	Stimulation procedure (procedure)	Procedure to arouse the body or any of its parts or organs to increase functional activity
32485007	P0-10000	Hospital admission (procedure)	Performance of the steps necessary to admit a patient to a hospital
58000006	P0-20000	Patient discharge (procedure)	Performance of the steps necessary to discharge a patient from a location of care delivery
107724000	P0-20301	Patient transfer (procedure)	Performance of the steps necessary to transfer a patient between locations of care delivery
633004	P0-80000	Chart review by physician (procedure)	A chart evaluation performed by a physician
107727007	P0-80001	Chart related administrative procedure (procedure)	An administrative procedure that involves a medical record chart
107728002	P0-80002	Chart evaluation by healthcare professional (procedure)	An evaluation of a medical chart by a health care professional
86078004	P0-80500	Quality of care procedure (procedure)	A procedure that assesses the quality of health care service delivery
50659003	P0-80600	Medical audit procedure (procedure)	A quality of care determination performed retrospectively
16817006	P0-80660	Medical service audit (procedure)	A medical audit of direct care providers

CONCEPTID	*SNOMEDID*	*FULLY SPECIFIED NAME*	DEFINITION
15367004	P0-80690	Ancillary service audit (procedure)	A medical audit of ancillary services (such as physical therapy, dietary)
15807005	P0-806A0	Financial audit (procedure)	A financial procedure that assesses a financial situation
57430006	P0-80800	Chart evaluation by non-healthcare professional (procedure)	A chart-related administrative procedure that checks a chart for completion and accuracy and conformance to chart policy
54455007	P0-80820	Chart review, verification of charges (procedure)	A financial audit to review and/or verify charges
29064005	P0-80850	Chart opening (procedure)	A chart-related administrative procedure that involves opening the chart
6035001	P0-80860	Chart abstracting (procedure)	A chart related administrative procedure that involves abstracting information from the chart
42965003	P0-80890	Chart completion by medical records (procedure)	A chart related administrative procedure done by the medical records department
118662001	P0-90003	Medicolegal procedure (procedure)	An administrative legal procedure
118629009	P0-A0201	Functional training (procedure)	Procedure aimed at enhancing functioning, frequently includes repetition of actions to develop, re-create or maintain physiological/cognitive processes.
118635009	P1-00031	Revision (procedure)	Repeating a prior procedure to correct or improve the results
34896006	P1-01000	Incision (procedure)	Making a cut in something
76145000	P1-01001	Exploratory incision (procedure)	An incision done for the purpose of performing an exploration
122458006	P1-01002	Exploration procedure (procedure)	An observation of the body or a body part done by inspection and/or palpation
122459003	P1-01003	Dissection procedure (procedure)	A separation of different structures along natural cleavage lines by dividing the connective tissue framework
19484001	P1-01008	Decompressive incision (procedure)	An incision that relieves abnormal pressure on a structure

CONCEPTID	*SNOMEDID*	*FULLY SPECIFIED NAME*	DEFINITION
122460008	P1-01009	Reexploration procedure (procedure)	A repeated exploration
122461007	P1-01012	Evacuation procedure (procedure)	Removal of the contents of a body cavity or container
122462000	P1-01013	Drainage procedure (procedure)	Evacuation of liquid contents by gravity
70302008	P1-01020	Transection (procedure)	A division made transversely across a long axis
118630004	P1-01027	Division (procedure)	An incision that separates something into two or more parts
387651008	P1-010A1	Exploration with a probe (procedure)	An exploration done using a probe
85921004	P1-01100	Puncture procedure (procedure)	A procedure done by piercing or penetrating with a pointed object or instrument
86088003	P1-01130	Centesis (procedure)	A puncture into a space with an aspiration of that space
65801008	P1-03000	Excision (procedure)	Removal done with a cutting instrument
79095000	P1-03002	Complete excision of organ (procedure)	Complete excision and removal of an entire body organ
118292001	P1-03003	Removal (procedure)	To take something off or out, to get rid of, to eliminate
20418004	P1-03004	Wedge resection (procedure)	Excision of a wedge-shaped piece of tissue (often but not necessarily for diagnostic examination)
118636005	P1-0300B	Expulsion (procedure)	Evacuation using positive pressure
81723002	P1-03030	Amputation (procedure)	Excision of normal topography
15440009	P1-03038	Disarticulation (procedure)	Amputation through a joint without cutting of bone
14509009	P1-03053	Evisceration (procedure)	Radical excision of tissues and organs of a body cavity

CONCEPTID	*SNOMEDID*	*FULLY SPECIFIED NAME*	DEFINITION
39250009	P1-03056	Enucleation (procedure)	A removal of an anatomic or pathologic structure in entirety without breakage
86273004	P1-03100	Biopsy (procedure)	Removal of tissue for diagnostic examination
8889005	P1-03101	Excisional biopsy (procedure)	Biopsy that removes an entire lesion, with or without surrounding tissue
70871006	P1-03102	Incisional biopsy (procedure)	Biopsy that involves incision and removal of part of a lesion or organ, rather than excision of the entire lesion or organ
119283008	P1-03103	Open biopsy (procedure)	Biopsy by open approach, as opposed to percutaneous or endoscopic methods
14766002	P1-03130	Aspiration (procedure)	Extraction using negative pressure
36777000	P1-03140	Debridement (procedure)	Removal of devitalized tissue
68688001	P1-03150	Curettage (procedure)	Scraping done with a curette
29923002	P1-03153	Shaving (procedure)	A scraping away of thin sections
56757003	P1-03154	Scraping (procedure)	Removal from a surface by repeated strokes of an edged instrument
107733003	P1-04FFF	Introduction (procedure)	Introduction of object AND/OR substance into or onto body, including injection, implantation, infusion, perfusion, transfusion, irrigation, instillation, insertion, placement, replacement, packing, intubation, catheterization , cannulation
59108006	P1-05000	Injection (procedure)	Administration using positive pressure and a needle or other equipment to drive a substance into the body
14792009	P1-05015	Tattooing (procedure)	An injection of indelible pigments
119268003	P1-05025	Inflation (procedure)	Insufflation of a hollow organ or body cavity with gas, causing it to distend or swell
36576007	P1-05030	Infusion (procedure)	An injection that is continuous
67889009	P1-05050	Irrigation (procedure)	Administration that washes with a stream of liquid

CONCEPTID	*SNOMEDID*	*FULLY SPECIFIED NAME*	DEFINITION
68641000	P1-05060	Insufflation (procedure)	An injection of a gas or powder into a body cavity by positive pressure
55870005	P1-05070	Instillation (procedure)	Administration of a liquid, drop by drop, into or onto the body
122463005	P1-05080	Embolization procedure (procedure)	An injection of some substance into the circulation to occlude vessels, either to arrest or prevent hemorrhaging or to devitalize a structure or organ by occluding its blood supply
71861002	P1-05500	Implantation (procedure)	Introduction of a non biologic device
3137001	P1-05501	Reimplantation (procedure)	Implantation that is being revised
52765003	P1-05530	Intubation (procedure)	An insertion of a tubular device into a canal, hollow organ, or cavity
4365001	P1-08000	Surgical repair (procedure)	Restoring, to the extent possible, the natural anatomical structure
122464004	P1-08005	Augmentation procedure (procedure)	Procedure to increase the size, shape, or volume of a body structure
59719002	P1-08060	Exteriorization (procedure)	To expose the inner surface of a structure to the external surface of the body
118627006	P1-08061	Marsupialization (procedure)	A construction of a pouch, achieved by resecting the anterior wall of a cyst or other enclosed cavity and suturing the cut edges of the remaining wall to adjacent edges of skin
112695004	P1-08400	Reparative closure (procedure)	A repair that unites structures
50015006	P1-08413	Closure by staple (procedure)	A closure done by stapling
70751009	P1-08420	Ligation (procedure)	To bind with a ligature
56275003	P1-08421	Suture ligation (procedure)	A ligation where the surgical suture serves as a ligature
1431002	P1-08460	Fixation (procedure)	The act or operation of holding, suturing, or fastening in a fixed position
122868007	P1-08490	Staple implantation procedure (procedure)	An implantation of a staple

CONCEPTID	*SNOMEDID*	*FULLY SPECIFIED NAME*	DEFINITION
122465003	P1-08601	Reconstruction procedure (procedure)	A reparative construction that builds or rebuilds a structure that should normally be present
78817002	P1-08610	Construction of anastomosis (procedure)	A construction of an opening between two hollow structures, organs, or spaces, be they real or artificial
88834003	P1-08611	Construction of shunt (procedure)	A construction of an alternate route of passage of a bodily substance
75506009	P1-08612	Construction of stoma (procedure)	A construction of an abnormal passage between a cavity or hollow organ and the surface of the body
4116001	P1-08613	Construction of window (procedure)	A construction of openings or fenestrae
118626002	P1-08617	Construction of interposition anastomosis (procedure)	An anastomosis that places a tubular structure between the cut ends of a previously contiguous tubular structure
87193006	P1-08700	Fusion-stabilization and immobilization (procedure)	A fixation that joins together two body parts, rendering them immobile with respect to each other
122501008	P1-08702	Fusion procedure (procedure)	Procedure to cause two adjacent structures to be structurally joined together
122502001	P1-08703	Anchoring procedure (procedure)	Procedure to fix a mobile or flexible structure to a rigid or inflexible structure
82254000	P1-08710	Refixation (procedure)	A fixation that is being revised
64597002	P1-0C000	Destructive procedure (procedure)	Eradicating all or a portion of a body part
9667001	P1-0C002	Surgical avulsion (procedure)	A removal done by tearing away or forcible separation
2677003	P1-0C015	Stripping (procedure)	A removal done by peeling, often using a stripper
119266004	P1-0C025	Coagulation (procedure)	A destruction of tissue by means that results in condensation of protein material
119271006	P1-0C027	Obliteration (procedure)	A destruction of a natural space or lumen by induced fibrosis or inflammation

CONCEPTID	*SNOMEDID*	*FULLY SPECIFIED NAME*	DEFINITION
27411008	P1-0C080	Cauterization	A destruction of tissue by burning or searing with a thermal instrument, an electric *current*, or a caustic substance
43802008	P1-0C200	Thermocautery (procedure)	A cauterization done with thermal energy
60726007	P1-0C400	Crushing (procedure)	A destruction done by injurious pressure. Note that this pressure can be mechanical, as in squeezing between two hard bodies, or can be a pressure wave, as is used to crush internal stones.
82413005	P1-0C410	Litholapaxy (procedure)	A crushing of calculi (stone).
5845006	P1-0C430	Emulsification procedure (procedure)	A destruction achieved by turning a solid into an emulsion
64874008	P1-0C504	Chemodenervation (procedure)	A denervation done using chemicals
3324009	P1-0C620	Laser beam photocoagulation (procedure)	A photocoagulation using a laser beam
77465005	P1-0D000	Transplantation (procedure)	To move body tissue or cells from donor site to recipient site
53088000	P1-0D010	Autogenous transplantation (procedure)	A transplantation where the donor and recipient spots are part of the same organism
27782009	P1-0D012	Syngeneic transplantation (procedure)	A transplantation where the donor and recipient spots are part of genetically identical organisms
50223000	P1-0D016	Allogeneic transplantation (procedure)	A transplantation where the donor and recipient spots are from antigenically distinct individuals of the same species
48537004	P1-0D100	Bypass graft (procedure)	A construction of a shunt using either biologic or synthetic material
75152009	P1-0D200	Transposition procedure (procedure)	An autogenous transplantation that does not entirely sever the topographic object from the donor spot, at least until it is united at the recipient spot
19207007	P1-0E000	Manipulation (procedure)	Skilled dextrous action of the hands directly applied to a body part

CONCEPTID	*SNOMEDID*	*FULLY SPECIFIED NAME*	DEFINITION
74923002	P1-0E100	Mobilization (procedure)	A procedure that mobilizes or frees up an abnormally fixed structure
66391000	P1-0E150	Traction (procedure)	The act of exerting a pulling force
112696003	P1-0E200	Manual reduction (procedure)	A repair done via manipulation
62972009	P1-0E300	Extraction (procedure)	Removal done by pulling
10012005	P1-0E350	*Expression* (procedure)	An expulsion done by manipulation
62057008	P1-0E410	Dilation and stretching (procedure)	A dilation and a stretching
122546009	P1-0E411	Stretching procedure (procedure)	Enlarging or distending a structure, increasing its internal wall stress
2802005	P1-0E420	Manual dilation and stretching (procedure)	A dilation and stretching done by manipulation
9421007	P1-0E450	Bougienage (procedure)	A dilation done with a bougie
122467006	P1-0E501	Fitting procedure (procedure)	A measurement or adjustment of a device or biologic material to the right shape or size so as to conform correctly when introduced or transplanted
118659004	P1-10887	Tenosuspension (procedure)	Procedure to anchor a tendon to act as a suspensory ligament
7108004	P1-10C04	Osteoclasis (procedure)	A destruction that purposefully results in a fracture of bone
81099000	P1-13860	Cervical arthrodesis (procedure)	The stiffening of one or more cervical joints by operative means
70502009	P1-17A45	Pollicization of a digit (procedure)	The act of making a thumb out of a digit (finger or toe)
46809006	P1-19376	Ostectomy, excision of tarsal coalition (procedure)	Excision of the fibrous, cartilaginous, or bony fusion of two or more of the tarsal bones
359593004	P1-19A3E	Midtarsal fusion (procedure)	Arthrodesis of one or more of the tarsal joints
32350008	P1-41D60	Intermediate delay of small flap at scalp (procedure)	Delayed transfer flap-a flap graft that is partially raised from the donor bed to permit collateral circulation of the pedicle

CONCEPTID	*SNOMEDID*	*FULLY SPECIFIED NAME*	DEFINITION
57554003	P1-48120	Periprosthetic capsulotomy of breast (procedure)	Division of a fibrous capsule surrounding a prosthetic breast implant
90991008	P1-48303	Periprosthetic capsulectomy of breast (procedure)	Excision of a fibrous capsule surrounding a prosthetic breast implant
85768003	P1-78331	Excision of median bar of prostate by transurethral approach (procedure)	Median bar-a fibrotic structure across the neck of the prostate causing urethral obstruction
77066003	P1-79370	Excision of hydatid of Morgagni in male (procedure)	Excision of appendix of testis-vestige of Mullerian duct
28644007	P1-82308	Vaginal enterocelectomy (procedure)	Excision of an enterocele, a posterior vaginal hernia
32998005	P1-8280B	Latzko operation on vagina (procedure)	Repair of vesicovaginal fistula
60753006	P1-82826	McIndoe operation for construction of vagina (procedure)	Construction of an artificial vagina consisting of a mold covered with a split-thickness skin graft
64853007	P1-82827	Williams-Richardson operation for construction of vagina (procedure)	A vulvovaginoplasty procedure described by Williams to create a vaginal canal
174000	P1-82833	Harrison-Richardson operation on vagina (procedure)	Vaginopexy according to Williams and Richardson is an abdominal colposuspension by strips from external oblique
40587007	P1-82909	Pereyra procedure including anterior colporrhaphy (procedure)	Pereyra procedure: Needle suspension and suture of bladder neck for stress incontinence
112928008	P1-86E22	Crede maneuver (procedure)	A method of external massage of the uterus to promote delivery of the placenta
57551006	P1-A4824	Iridotasis (procedure)	Stretching of the iris to increase the outflow of aqueous from the eye in glaucoma patients
3700004	P1-A5344	Erysiphake extraction of cataract by intracapsular approach (procedure)	Intracapsular extraction of cataract using an erysiphake, an instrument used to aspirate a cataract
54305003	P1-A5820	Lens couching procedure (procedure)	Obsolete procedure involving displacement of lens into vitreous for treatment of cataract

CONCEPTID	*SNOMEDID*	*FULLY SPECIFIED NAME*	DEFINITION
84100007	P2-01000	History taking (procedure)	A clinically oriented interview of a patient or someone familiar with the patient
32166003	P2-01060	History taking, self-administered, questionnaire (procedure)	A history taken by a self-administered questionnaire
5880005	P2-01400	Physical examination procedure (procedure)	An observation of the body or a body part using one of the five human senses (e.g., inspection, palpation, percussion, auscultation)
32750006	P2-01500	Inspection (procedure)	An exploration using the sense of sight, done with the eyes
113011001	P2-01510	Palpation (procedure)	An exploration using the sense of touch
75180006	P2-01550	Percussion (procedure)	A listening of the sounds produced in response to tapping the body surface
37931006	P2-01560	Auscultation (procedure)	A listening to spontaneously generated body sounds
103741002	P2-01570	Optical transillumination (procedure)	An inspection by the passage of light through tissues or a body cavity
16076005	P2-08000	Prescription (procedure)	A legal *order* to dispense and possibly prepare a substance or physical object
1366004	P2-22100	Inhalation therapy procedure (procedure)	An administration into the respiratory tract by inspiration
4764004	P2-22223	Jet ventilation procedure (procedure)	Jet ventilation phasically directs a high-velocity jet of humidified gas into the endotracheal tube at rapid frequencies, entraining additional fresh gas during insufflation
40617009	P2-22902	Artificial respiration (procedure)	An assistance of respiration
11140008	P2-22905	Respiratory assist, manual (regime/therapy)	An artificial respiration done manually
23426006	P2-25010	Measurement of respiratory function (procedure)	A procedure on the respiratory tract that observes pulmonary function
76572000	P2-25250	Measurement of lung volume (procedure)	A pulmonary function test that measures lung volumes

CONCEPTID	*SNOMEDID*	*FULLY SPECIFIED NAME*	DEFINITION
9134004	P2-30600	Ballistocardiography (procedure)	Obsolete method for determining cardiac output by measuring recoil of body due to cardiac contraction
91480001	P2-40300	Iontophoresis procedure (regime/therapy)	An administration into the tissues of an ionic substance by means of an electric *current*
225426007	P2-4510F	Administration of therapeutic substance (procedure)	Introduction of a substance to the body
5447007	P2-68000	Transfusion (procedure)	An infusion of blood or blood product
239332003	P3-6018D	Percutaneous test for allergy (procedure)	An immune system procedure that observes for evidence of hypersensitivity
252512005	P3-60197	In vivo test of hypersensitivity (procedure)	An immune system procedure that observes for evidence of hypersensitivity
118640001	P5-C0900	Radioimmunotherapy (procedure)	Radiation therapy using radiolabelled antibodies
12894003	P7-00040	Functional assessment (procedure)	An evaluation of the performance of an organ, organ system, or body part
71937005	P7-10000	Physiatric manipulation (regime/therapy)	A manipulation done by a physiatrist
16992002	P7-11000	Osteopathic manipulation (procedure)	A manipulation done by an osteopath
46947000	P7-12000	Chiropractic manipulation (procedure)	A manipulation done by a chiropractor
15420002	P8-80770	Reline lower partial denture, laboratory (procedure)	Refitting a denture by replacing the denture base while keeping the occlusal *relationship* of the teeth the same
44764005	P8-85460	Crown, porcelain fused to noble metal (procedure)	Crowning preparation and covering of the natural crown of a tooth with a veneer consisting of a metal, plastic resin, or porcelain or combinations
255482005	R-40491	Left upper segment (*qualifier* value)	At location of left laterality and superior
255496004	R-4049E	Right lower segment (*qualifier* value)	At location (or vessel branch as in pulmonary vein) of right laterality and inferior

CONCEPTID	*SNOMEDID*	*FULLY SPECIFIED NAME*	DEFINITION
255499006	R-404A0	Right upper segment (*qualifier* value)	At location (or vessel branch as in pulmonary vein) of right laterality and superior
246464006	R-42019	Function (observable entity)	Any function or property that is not mainly morphologic or structural, including both measurable and observable features and physiologic actions
264068005	R-4214B	Left lower segment (*qualifier* value)	At location of left laterality and inferior
312004007	R-42E61	Retrograde direction (*qualifier* value)	The state of reverse blood flow through a valve or orifice
125681006	S-11033	Single person (finding)	Not currently married
33553000	S-11040	Widowed (finding)	An unmarried person whose spouse has died
44667005	T-02660	Skin structure of hand, including finger (body structure)	Skin region including some skin of finger AND some additional non finger skin
9385004	T-03660	Subcutaneous tissue structure of hand, including finger (body structure)	Subcutaneous tissue including some tissue of finger AND some additional non finger tissue
84157002	T-11039	Structure of epiphyseal line (body structure)	The location of the epiphyseal growth plate subsequent to its ossification
108372004	T-12762	Entire tarsal bone (body structure)	Any bone that is part of the tarsus
91739007	T-13007	Endomysium (body structure)	Fine connective tissue sheath around a muscle fiber
122504000	T-15624	Manubriosternal synostosis (body structure)	The connection between the manubrium and sternum that has progressed from a symphysis to bony *union* (synostosis)
122497003	T-17441	Flexor tendon of wrist (body structure)	Tendons involved in flexing the wrist joint, excluding flexor tendons that pass through the wrist that flex the fingers
6553002	T-21370	Inferior nasal turbinate structure (body structure)	Shell-shaped structure of lateral inferior nasal cavity, including bone and covering mucous membrane

CONCEPTID	*SNOMEDID*	*FULLY SPECIFIED NAME*	DEFINITION
60962000	T-21380	Middle nasal turbinate structure (body structure)	Shell-shaped structure of lateral middle nasal cavity, including bone and covering mucous membrane
65289004	T-21390	Superior nasal turbinate structure (body structure)	Shell-shaped structure of lateral superior nasal cavity, including bone and covering mucous membrane
33415007	T-21391	Supreme nasal turbinate structure (body structure)	Shell-shaped structure of lateral nasal cavity above the superior nasal turbinate, including bone and covering mucous membrane
3377004	T-21420	Structure of agger nasi (body structure)	Ridge on the lateral internal nasal wall due to the ethmoidal crest of the maxilla
81040000	T-44000	Pulmonary artery structure (body structure)	Includes pulmonary trunk, left and right main pulmonary arteries, and all their branches
128261004	T-44004	Mediastinal pulmonary artery (body structure)	Includes pulmonary trunk and left and right main pulmonary arteries
69105007	T-45010	Carotid artery structure (body structure)	One of the common carotid, internal carotid, or external carotid arteries
38809004	T-48300	Structure of vein of thorax (body structure)	Vein located within the thorax
91539005	T-48501	Structure of right pulmonary vein (body structure)	One of the great vessels draining venous blood from the right lung
27706005	T-48502	Structure of left pulmonary vein (body structure)	One of the great vessels draining venous blood from the left lung
122972007	T-48581	Pulmonary venous structure (body structure)	Any vein draining the lungs, including pulmonary veins proper and their tributaries
128553008	T-49215	Structure of antecubital vein (body structure)	A vein located in the antecubital fossa
51289009	T-50100	Digestive tract structure (body structure)	Entire digestive tract including mouth,esophagus , stomach, and intestines
122865005	T-50101	Gastrointestinal tract structure (body structure)	Esophagus , stomach, small intestine, and large intestine together as a single entity
62834003	T-50110	Upper gastrointestinal tract structure (body structure)	Esophagus , stomach, and duodenum

CONCEPTID	*SNOMEDID*	*FULLY SPECIFIED NAME*	DEFINITION
5668004	T-50120	Lower gastrointestinal tract structure (body structure)	Jejunum, ileum, colon, rectum, and anal canal
34810001	T-51010	Structure of vestibule of mouth (body structure)	The part of the oral cavity external to gums and teeth and internal to cheeks and lips
71617008	T-51020	Structure of oral cavity proper (body structure)	The part of the oral cavity internal to the teeth and bounded posteriorly by the palatoglossal arch
47975008	T-53130	Structure of root of tongue (body structure)	The part of the tongue that is on the floor of the mouth, and is not covered by mucous membrane
47975008	T-53130	Structure of root of tongue (body structure)	The pharyngeal part of the tongue, forming the anterior wall of the oropharynx
3100007	T-54026	Clinical crown of tooth (body structure)	Portion of tooth exposed above gums, the part above the clinical root
11326003	T-54061	Structure of coronal pulp of tooth (body structure)	Part of the pulp of tooth that is within the crown portion of the pulp cavity
38954004	T-545A1	Carnassial tooth structure (body structure)	Tooth adapted to shear flesh
68028003	T-56500	Crop (body structure)	Dilated part of esophagus for food storage in birds
118652008	T-58049	Crypt of Lieberkühn (body structure)	Denotes the pits or crypts, but not the glands that lie beneath them
3789000	T-58140	Enteroendocrine cell (cell)	Endocrine cell of the gut
128180007	T-59261	Pelvic appendix (body structure)	Appendix which is oriented posteriorly and inferiorly in the pelvic cavity
122489005	T-70001	Urinary system structure (body structure)	Organs of urine formation and secretion
49549006	T-A0050	Visual system structure (body structure)	The eye, ocular adnexa, afferent visual pathways, efferent visual pathways, and pupil innervation pathways

CONCEPTID	*SNOMEDID*	*FULLY SPECIFIED NAME*	DEFINITION
1231004	T-A1110	Meninges structure (body structure)	The three membranes that surround the brain and spinal cord, consisting of the dura mater, arachnoid, and pia mater. The pia and arachnoid in combination are referred to as the leptomeninges.
35764002	T-A1600	Brain ventricle structure (body structure)	The four ventricles of the brain, including the two lateral ventricles, the third ventricle, and the fourth ventricle
35328001	T-A9814	Structure of hepatic plexus (body structure)	An unpaired autonomic plexus that is part of the celiac plexus that lies on the hepatic artery and its branches in the liver
82200002	T-A9815	Structure of splenic plexus (body structure)	An autonomic plexus that is a subdivision of the celiac plexus that accompanies the splenic artery
64157005	T-A9816	Structure of gastric plexus (body structure)	Autonomic plexi that are part of the celiac plexus that lies on the greater and lessor curvatures of the stomach
2504000	T-A9817	Structure of pancreatic plexus (body structure)	An autonomic plexus that is a subdivision of the celiac plexus and accompanies the pancreatic arteries
49412003	T-A9818	Adrenal plexus (body structure)	An autonomic plexus that is a subdivision of the celiac plexus that accompanies the adrenal artery
48563009	T-A9819	Structure of renal plexus (body structure)	An autonomic plexus that is a subdivision of the celiac plexus that accompanies the renal artery
21684005	T-A981A	Structure of ureteral plexus (body structure)	An autonomic plexus that is derived from the celiac plexus, more specifically renal and hypogastric plexi, that accompanies the ureteric artery to the ureter
86064000	T-A981B	Structure of testicular plexus (body structure)	An autonomic plexus that is a subdivision of the aortic plexus, or derived from it, that accompanies the testicular artery
15398002	T-A981C	Structure of ovarian plexus (body structure)	An autonomic plexus that is a subdivision of the aortic plexus, or derived from it, that accompanies the ovarian artery

CONCEPTID	*SNOMEDID*	*FULLY SPECIFIED NAME*	DEFINITION
53224005	T-A9820	Superior mesenteric plexus structure (body structure)	An autonomic plexus that branches from the aortic plexus, that sends nerves to intestines and with the vagus forms subserous, myenteric, and submucous plexus
4150005	T-A9821	Celiac nervous plexus structure (body structure)	An autonomic plexus that is a superior subdivision of the aortic plexus that runs anterior to the aorta at the level of the celiac trunk T12; contains celiac ganglia and most visceral afferents pass though it
74636004	T-A9822	Structure of aorticorenal ganglia (body structure)	A part of the celiac ganglion that is semidetached and contains sympathetic neurons that innervate the kidney
79869003	T-A9830	Inferior mesenteric plexus structure (body structure)	An autonomic plexus that branches from the aortic plexus, that sends nerves to descending colon, sigmoid, and rectum along the path of the inferior mesenteric artery
24981001	T-A9831	Structure of superior rectal plexus (body structure)	An autonomic plexus that branches from the inferior mesenteric plexus; accompanies superior rectal artery to rectum
69854003	T-A9840	Structure of Auerbach's plexus (body structure)	A subdivision of the enteric plexus that lies within the tunica muscularis of the intestinal tract
8595004	T-A9841	Structure of Meissner's plexus (body structure)	Part of the enteric plexus situated in the intestinal submucosa
68305005	T-A9842	Structure of iliac plexus (body structure)	Autonomic plexus derived from the aortic plexus, accompanying iliac arteries
34109009	T-A9843	Structure of femoral plexus (body structure)	An autonomic plexus accompanying the femoral artery and derived from the iliac plexus
47881004	T-A9850	Inferior hypogastric plexus structure (body structure)	An autonomic plexus formed by the junction of the hypogastric nerve and the splanchnic nerve on each side; supplies pelvic viscera
53892006	T-A9851	Structure of hypogastric nerves (body structure)	Nerves in the pelvis that connect the superior hypogastric plexus to the inferior hypogastric plexus
13559005	T-A9860	Structure of superior hypogastric plexus (body structure)	A continuation of the aortic plexus that leads to the right and left hypogastric nerves

CONCEPTID	*SNOMEDID*	*FULLY SPECIFIED NAME*	DEFINITION
26701009	T-A9861	Structure of pelvic ganglia (body structure)	Sympathetic and parasympathetic ganglia within pelvic plexi
17630002	T-A9862	Medial rectal plexus (body structure)	A subdivision of inferior hypogastric plexus, maybe derived from it, that supplies nerves to the rectum
69079004	T-A9864	Prostatic sympathetic plexus (body structure)	Autonomic plexus that runs through the capsule of the prostate and is derived from the inferior hypogastric plexus, and supplies the cavernous nerves of the erectile tissue of penis
4905007	T-A9866	Structure of uterovaginal plexus (body structure)	Autonomic plexus with ganglia derived from inferior hypogastric plexus, that supplies uterus, vagina, ovary, erectile tissue of vestibule, and urethra
2044003	T-A9867	Structure of vaginal nerves (body structure)	Nerves running from uterovaginal plexus to vagina that are both sympathetic and parasympathetic
81036009	T-A9868	Structure of vesical plexus (body structure)	An autonomic plexus derived from inferior hypogastric plexus to supply sympathetic nerve fibers to urinary bladder, ureter, ductus deferens, and seminal vesicle
36364006	T-A9869	Structure of cavernous nerves of penis (body structure)	Two nerves that supply sympathetic and parasympathetic innervation to vascular structures of corpus cavernosum, stimulating erection; derived from prostatic plexus
43671000	T-A986A	Structure of cavernous nerves of clitoris (body structure)	Nerves that supply sympathetic and parasympathetic innervation to erectile tissue of clitoris; derived from uterovaginal plexus
56190005	T-AA003	Structure of external axis of eyeball (body structure)	A line segment connecting anterior pole of cornea to posterior pole of sclera
86588008	T-AA008	Structure of muscular fascia of eyeball (body structure)	Fascia enclosing the extrocular muscles
24736005	T-AA00B	Structure of soft tissues of orbit (body structure)	Soft tissues enclosed within orbital region
26386000	T-AA079	Vitreous cavity structure (body structure)	The eye cavity that contains the vitreous

CONCEPTID	*SNOMEDID*	*FULLY SPECIFIED NAME*	DEFINITION
47538007	T-AA081	Vitreous body structure (body structure)	Composite structure of hyaluronic acid gel within a stromal network of collagen fibrils
19359003	T-AA083	Hyaloid canal structure (body structure)	A canal that runs from optic disc to lens that contains the hyaloid artery in the fetus
30110007	T-AA084	Structure of hyaloid fossa (body structure)	A depression of the anterior surface of the vitreous body where the lens fits
76729009	T-AA100	Fibrous tunic of eye structure (body structure)	Outer coating of eyeball; has parts cornea and sclera
27941003	T-AA101	Structure of anulus tendineus communis (body structure)	Orbital structure; a fibrous ring that is the origin of the rectus muscles
31808004	T-C4310	Entire intrapulmonary lymph node (body structure)	Lymph node within the lung
65690001	T-C4340	Structure of paratracheal lymph node (body structure)	Lymph node along the trachea; may be in either the thorax or the neck
11899006	T-C4365	Structure of esophageal lymph node (body structure)	Lymph node along the esophagus , may be in either the mediastinum or the neck
113343008	T-D0060	Body organ structure (body structure)	An anatomical structure that consists of the maximal set of organ parts so connected to one another that together they constitute a self-contained unit of macroscopic anatomy, distinct both morphologically and functionally from other such units. Together with other organs, an organ constitutes an organ system or a body part. An organ is divisible into organ parts but not organs (examples: femur, biceps, liver, heart, aorta, sciatic nerve, ovary).
118760003	T-D0066	Entire viscus (body structure)	A large organ in the thorax, abdomen, or pelvis
71905009	T-D1209	Infrapalpebral fold structure (body structure)	The sulcus (furrow) below the lower eyelid
122861001	T-D1217	Sublingual space (body structure)	A potential space in the floor of the mouth; the part of the submandibular space above the mylohyoid muscle
122498008	T-D1219	Submaxillary space structure (body structure)	A potential space of the floor of the mouth; part of the submandibular space below the mylohyoid muscle

CONCEPTID	*SNOMEDID*	*FULLY SPECIFIED NAME*	DEFINITION
122499000	T-D121A	Submental space (body structure)	A potential space of the floor of the mouth; the medial part of the submaxillary space
122500009	T-D121B	Masticator space (body structure)	A potential space containing the pterygoid and masseter muscles
49928004	T-D1602	Structure of anterior portion of neck (body structure)	Neck anterior to the vertebral column, including pharynx, larynx, and anterior surface
16982005	T-D2220	Shoulder region structure (body structure)	The body part defined by the shoulder joint and its surrounding structures
29836001	T-D2500	Hip region structure (body structure)	The body part defined by the hip joint and surrounding structures, including the region from the iliac crest to the thigh
8389004	T-D6242	Structure of obstetrical canal (body structure)	The birth canal; consists of uterine cervix, vagina, and vulva
53120007	T-D8000	Upper limb structure (body structure)	Upper extremity, including shoulder, arm, forearm, wrist, and hand
40983000	T-D8200	Upper arm structure (body structure)	Upper extremity between shoulder and elbow
76248009	T-D8300	Entire elbow region (body structure)	The body part defined by the elbow joint and surrounding structures
8205005	T-D8600	Wrist region structure (body structure)	The body part defined by the wrist joint and surrounding structures
61685007	T-D9000	Lower limb structure (body structure)	Lower extremity, including hip, thigh, leg, ankle, and foot
83038001	T-D9040	Femoral region structure (body structure)	Anterior region of the thigh
72696002	T-D9200	Knee region structure (body structure)	The body part defined by the knee joint and its surrounding structures
30021000	T-D9400	Lower leg structure (body structure)	Lower extremity from knee to ankle
54134001	T-F1200	Extraembryonic membranes structure (body structure)	The membranes which surround the embryo but are not involved in formation of the embryo itself

3.2.13 Enumerated values

Table 209: Enumerated Values for Component Status

Value	**Name**	***Description***	**Table and Field**		
			ConceptStatus	***DescriptionStatus***	***ComponentHistory Status***
0	*Current*	*active component* in *current* use	Y	Y	Y
1	*Retired*	Withdrawn without a specified *reason*.	Y	Y	Y
2	Duplicate	Withdrawn from *active* use because it duplicates another *component*.	Y	Y	Y
3	Outdated	Withdrawn from *active* use because it is no longer recognized as a valid clinical *concept* / *term*.	Y	Y	Y
4	Ambiguous	*Concept* withdrawn from *active* use because it is inherently ambiguous.	Y		Y
5	Erroneous	Withdrawn from *active* use as it contains an error. A corrected but otherwise similar *Concept* has been added to replace it.	Y	Y	Y
6	Limited	*Inactive Concept* of limited clinical value as it is based on a classification *concept* or an administrative definition. Also applies to *active Descriptions* associated with a limited *Concept*.	Y	Y	Y
7	Inappropriate	*Description* withdrawn as this *term* should not refer to this *Concept*.		Y	Y
8	*Concept Retired*	*Description* contains a valid *term* used to describe an *inactive Concept*.		Y	Y
9	Implied	*Relationship retired* but still true as implied by other *Relationships*.			Y
10	*Moved elsewhere*	The *Concept* is no longer maintained by the organization responsible for the *namespace* of this *ConceptId*. The *Concept* has been moved to another *namespace* indicated by an associated | Moved to |*Relationship*.	Y	Y	Y

Value	Name	*Description*	Table and Field		
			ConceptStatus	***DescriptionStatus***	***Component History Status***
11	*Pending move*	The organization responsible for the *namespace* of this *ConceptId* has requested that it be moved to another *namespace* indicated by an associated | Moved to |*Relationship*. However, the move has not yet been confirmed by the recipient organization . *Concepts* with this *status* are still available for *active* use.	Y	Y	Y

Table 210: Enumerated Values in Descriptions Relationships and Subset Tables

Table and Field	Value	Name	*Description*
Descriptions *DescriptionStatus*	-	-	See *Table 209*
Descriptions *Description* Type	0	Unspecified	A *Description Subset* for a *language*, *dialect* or *realm* may assign this as a *Preferred Term* or *Synonym*
	1	Preferred	
	2	*Synonym*	
	3	FullySpecifiedName	
Relationships *CharacteristicType*	0	Defining	This *Relationship* represents a *defining characteristic* of the source *concept*. Hierarchical *Relationships* (e.g. " | is a | ") are also regarded as defining *Relationships*.
	1	*Qualifier*	
	2	Historical	
	3	Additional	
Relationships *Refinability*	0	Not refinable	Not refinable.
	1	Optional	
	2	Mandatory	

Table and Field	**Value**	**Name**	***Description***
Subsets *SubsetType*	1	*Language*	*Descriptions* appropriate to a specified *language* or *dialect*.
	2	*Realm Concept*	
	3	*Realm Descriptions*	
	4	*Realm Relationship* (for future use)	
	5	Context *Concept*	
	6	Context *Description*	
	7	*Navigation*	
	8	*Duplicate Terms Subset*	

Table 211: Enumerated Values in History and Cross Map Tables

Table and Field	**Value**	**Name**	***Description***
Component History Status	-	-	See *Table 209*
Component History Change Type	0	Added	The *component* was added to *SNOMED CT* in this *Release Version*.
	1	*Status* Change	
	2	Minor Change	
References Reference Type	1	Replaced by	Refers to a revised replacement for the *component*.
	2	Duplicated by	
	3	Similar to	
	4	Alternative	
	5	Moved To	
	6	Moved From	

Table and Field	Value	Name	***Description***
	7	Refers to *Concept*	
Cross Map Sets *MapSetType*	0	Unspecified	
	1	Single	
	2	Multiple	
	3	Choice	
	4	Flexible	

Table 212: Enumerated Values in Word Equivalents Table

Table and Field	Value	Name	***Description***
Word Equivalents *WordType*	0	unspecified	The default value.
	1	word form variant	
	2	*word equivalents*	
	3	abbreviation or acronym	
	4	equivalent phrase	
Word Equivalents *WordRole*	0	unspecified	The default value.
	1	general *qualifier*	
	2	topography	
	3	topography *qualifier*	
	4	object	
	5	action	
	6	unit of measure	

3.3 Release Format 2 Update Guide

3.3.1 Introduction

3.3.1.1 Purpose

The purpose of *RF2* is to provide a format that is flexible, unambiguous and useful. Its primary aim is to strengthen *SNOMED CT* by providing a format that is simple and stable, while enabling innovation through adaptations to cater for changing requirements.

This specification was developed by harmonizing proposals reviewed by the *IHTSDO* Enhanced *Release Format* Project Group, including:

- The Enhanced *Release Format* Specification (International Health Terminology Standards Development Organisation. *SNOMED Clinical Terms ® Enhanced Release Format Proposed Specification* , 21 June 2007).
- The *Reference Set* Specification (International Health Terminology Standards Development Organisation. *SNOMED Clinical Terms ® Reference Sets - Proposed Specification* , 31 July 2007).
- The Alternate *Release Format* proposed by NEHTA in coordination with their Australian Affiliates.

3.3.1.2 Who should read this guide?

The intended audience for this guide includes technical professionals who are involved in the development and/or implementation of healthcare information systems that use *SNOMED CT*.

For detailed technical guidance on the existing *Release Format*, please consult the *SNOMED CT* Technical *Reference* Guide (TRG) and *SNOMED CT* Technical Implementation Guide (TIG), as well as other applicable technical documentation described in the Associated Documentation table.

For technical guidance on using *Release Format 2*, please consult the "*SNOMED CT Release Format 2* - *Reference Set* Specifications" and the "*SNOMED CT Release Format 2* - Data Structures Specification" documents on the Collaborative site.

3.3.1.3 Associated Quality Measures

Although the definition of quality measures to monitor the implementation of this standard do not fall under the scope of this guide, they will be covered by the documentation covering the QA and *Release* process for the *Workbench*.

3.3.1.4 Summary of Changes

The *RF2* introduces a number of new *concepts* and capabilities. These are summarized below, and described in more detail later in this guide:

- Addition of an *Identifier file* to allow components to be identified by an arbitrary number of *Identifiers* from an arbitrary number of *Identifier* schemes;
- Addition of a module *Identifier* field to all components, enabling the source module in which each component is maintained to be identified, facilitating configuration management;
- Modified handling of the *language* and *dialect* properties of *descriptions*, for reduced complexity with increased utility;
- Introduction of *concept* enumerations making enumerations within *SNOMED CT* more easily extensible, self contained within the terminology (not dependent upon external documentation) and easily compatible across multiple *languages*;
- Addition of a *description logic* modifier *concept* enumeration to the *Relationship file* to represent different *Description Logic relationship types*, for example - some, all, all-some, not-some etc.

A general extensibility design pattern has also been introduced, which allows specification of a number of *Reference Set* formats, to meet different use cases. In *RF2*, *reference sets*:

- Result from the combination of generic *Reference Set* data structures, a design pattern and the application of domain *constraints* according to documented implementation guidelines;
- Use a machine readable model (called a *Reference Set* descriptor) that defines the extended information pertinent to a specific *Reference Set*;
- Make use of *concept* enumerations for representing optional information to enable machine-readability and increased extensibility;
- Apply the same history tracking and naming conventions as used elsewhere in *RF2*.

The *RF2* enhancements all contribute to greater flexibility and more explicit and comprehensive version control than *RF1*, and additionally introduce new features for configuration management. As a result, *RF2* is expected to accommodate evolving collaborative requirements without a need for further fundamental change in the foreseeable future. Since change to the *Release Format* causes difficulty and incurs cost to content developers, implementers and release centers alike, the *RF2* design is expected to result in long *term* savings as well as improvement in product functionality and quality.

3.3.1.5 Timescales for change

It should be noted that there is a difference between the release schedule of *RF1* / *RF2* in official IHSTDO-supported *International Releases*, and the release schedule of *RF1* / *RF2* in Member NRC releases. It is entirely possible that *RF1* will have a longer lifespan in Member NRC releases than in *IHTSDO International Releases*.

Actual timescales for *migration* of the *International release* to *RF2* are provided under separate notices, and have not been included in this guide as they are likely to follow a different review cycle.

3.3.2 Principles used in the design of RF2

The following principles were used to guide modifications made to the *Release Format*:

- Consistent history representation across all components and across all artifacts deemed in scope of the *Release Format*.
- Consistent identification of all components throughout their lifecycle and clear identification of all other artifacts in scope of the *Release Format*.
- Consistent representation of allowable values for component characteristics.
- Consistent means of extending component data structures to meet future requirements without modification to the existing table structures.
- Consistent non-centralized means of loosely coupled *Identifier* assignment for components and component characteristics.
- Consistent means of representing localizations and translations for all components.
- The data structures should assist implementers to consistently implement *SNOMED CT*. *Component* data structures ideally should not have to change to accommodate changes in editorial policies.
- Ideally component data structures should be simple, generic and flexible.
- Ideally component data structures should be self-contained, removing dependence on external artifacts .
- Dependencies between components should be explicitly stated and machine-readable. For example, it should be possible to express that a *reference set* released as part of an *extension* is dependent upon version X of the Acme *Extension* and version Y of the *SNOMED CT International Edition*.
- There must be a consistent means of identifying modules and their versions --including the *SNOMED CT International Release* itself.
- The *Release Format* should minimize the total effort of meeting requirements where possible by reuse of existing data structures.
- Metadata should be machine-readable.

- *Component* data structures that enable software reuse are preferred over data structures that require special development of parsers.
- It should be possible to produce from a *Release Format* an instance of that release in the immediately prior *Release Format*.
- Specifications should be based on requirements derived from use cases that describe the scope and environment of their intended use.
- *IHTSDO* specifications should provide a common global foundation that permits the development and maintenance of *SNOMED CT enabled applications* that are interoperable across national and organizational boundaries.
- Changes to *IHTSDO* specifications should only be made if the impact on implementation is considered to be proportionate to benefit. Such changes should be recorded.
- Changes to *IHTSDO* specifications should be evolutionary and should deliver incremental benefits to implementers with a minimum of disruption and re-engineering.
- The *SNOMED CT release format* and associated guidance should facilitate a consistent implementation for known use cases.
- The specifications that support implementation of use cases should be done in a way that doesn't limit the ability to realize other use cases within the scope of *SNOMED CT*.
- The *Release Format* is intended to be a distribution format and is not designed to be an implementation format.
- The *Release Format* should be designed to be consumed efficiently.

3.3.3 Rationale for moving from RF1 to RF2

3.3.3.1 Overview of Release Format 1

The current *SNOMED CT release format* has been in use since January 2002. During this period the generic and reusable aspects of the existing release structure have been a considerable strength.

Despite this success, there are a number of commonly accepted inconsistencies and limitations in the current *SNOMED CT distribution format*. This section gives a brief overview of the current *SNOMED CT distribution format*, and describes these limitations. For more details see the *Release Format 1 - Detailed specification*.

The current *RF1* format is summarized in *Figure 42*.

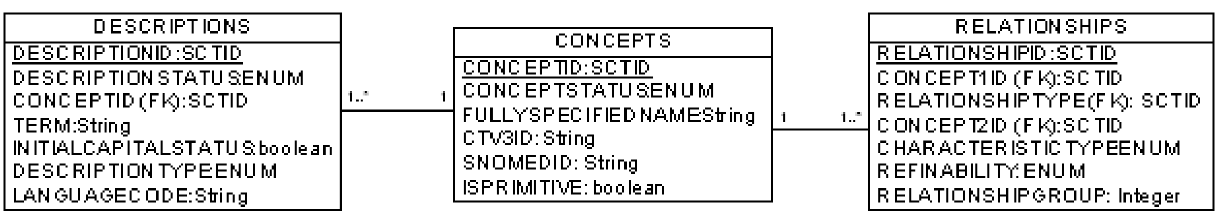


Figure 42: Release Format 1 Core Tables

Each *SNOMED CT concept* is held as a single row in the *CONCEPTS* file. Each *concept* may have one or more *descriptions* associated with it. Each *description* is held in a single row in the *Descriptions* file. Each *relationship*, from a source *concept* to a destination *concept*, is held as a single row in the *Relationships* file. The type of each *relationship* is defined by reference to a linkage *concept*, also held within the *CONCEPTS* file.

Separate file structures are also released to provide support for *maps* to other terminologies and coding systems, history tracking of *components* and *descriptions*, *subsets* of *SNOMED CT concepts* and other uses.

3.3.3.2 Drawbacks of Release Format 1

Inconsistencies and limitations in the current *Release Format* have led to a desire for a new *Release Format*. The following list briefly outlines these issues:

- Implicit semantics that must be inferred from external documentation that is not tightly coupled to changes in the terminology itself. Changes in the interpretation of data fields are not represented in the history of the data fields themselves.
- Pervasive use of *integer* enumerations within data fields, rather than using the self-referential means for representing symbolic constants provided for by the *SNOMED CT* terminology itself.
- No consistent and clearly defined mechanism for release centers , developers, implementers, and end users to extend the *RF1* data structures to meet unique and/or common needs not already provided for by the specifications and content of the *SNOMED CT International Release*.
- Inconsistent and unnecessarily complex data structures.
- Field overloading ""where one column represents multiple attributes (i.e. state and reason for inactivation).
- Inadequate Separation of Concerns, where data representation and data usage are often conflated, resulting in a difficulty in supporting software reuse and system evolution over time.
- Inconsistent and incomplete representation of terminology history resulting in a terminology that does not meet basic principles of configuration management and control.
- Inconsistent use of both enumerated values and *concepts* to enumerate values.
- Inconsistent naming and field ordering.
- *Term* length limited to 255 bytes and to plain text format.

Release Format 2 aims to address these issues.

3.3.3.3 Overview of Release Format 2

Release Format 2 consists of four primary files or tables. As in the current *SNOMED CT* format, all files:

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

The *core table* structure of *RF2* is similar to that of *Release Format 1*, although the fields within each of the *core files* are different. The *core files* within *RF2* consist of a *Concept file*, a *Description file*, and a *Relationship file*.

Each *SNOMED CT concept* is held as a single row in the *Concept file*. Each *concept* may have one or more *descriptions* associated with it. Each *description* is held in a single row in the *Description file*. Each *relationship*, from a source *concept* to a destination *concept*, is held as a single row in the *Relationship file*. The type of each *relationship* is defined by reference to a linkage *concept*, also held within the *Concept file*.

In addition to these files, an *Identifier file* has been added. This file holds one alternate *component Identifier* per row. Each alternate *Identifier* belongs to a particular *Identifier* scheme, and holds that scheme's *Identifier* for the *SNOMED CT Component* that it references. Within a scheme, each *Identifier* uniquely identifies a single *SNOMED CT component*.

The purpose of *RF2* is to provide a format that is flexible, adaptable, consistent, unambiguous and above all useful. Its primary aim is to strengthen *SNOMED CT* by providing a format that is simple yet flexible and powerful, allowing the format to remain constant, while allowing innovation and adaptation to changing requirements.

3.3.3.4 Backward compatibility

The proposed *RF2 Release Format* is backward compatible with the previous *SNOMED CT Release Format* (*RF1*), in the sense that all information contained within the current *Release Format* is represented, and legacy file formats can be derived from the new *Release Formats*. However, the *RF2* format contains functionality which is not supportable in the previous *Release Format*.

In order to achieve backward compatibility, the *RF2* may be transformed to create the previous distribution format. Additionally, *International releases* will be made available in both formats for a limited number of consecutive release cycles, for convenience.It is expected that *National Release* Centers will follow the same approach, and also release in dual format for a number of release cycles, unless there are specific reasons not to.

3.3.4 Details of Key Changes

The following subsections discuss details of the key changes between *RF2* and the previous *Release Format*.

3.3.4.1 Release Format structures and the abstract model

The following tables illustrate the correspondence between the fields of *Release Format 1* (*RF1*) and *Release Format 2* (*RF2*). *Release Format 2* fully represents the Logical Abstract Model for *SNOMED CT concept* and *derivatives*.

Release Format 1 represents a *snapshot view* which is less completely aligned with the logical model. As the tables illustrate all the information represented by *RF1* can be fully captured in the *RF2* representation. However, the reverse is not true. Therefore, some added functionality provided by *RF2* cannot be provided using the *RF1* data.

Table 213: Map between RF1 Concepts Table and RF2 Concept file

RF1 Concepts Table	***RF2 Concept File***
Concepts. *ConceptId*	*Concept*.id
<not supported> *RF1* releases contain a snap-shot view of the state of each *concepts* at the time of release.	*Concept*. *effectiveTime*

Concepts. *ConceptStatus*
• *active* :
  • 0 (*Current*);
  • 11 (*Pending Move*).
• *Inactive*:
  • All other values.
	Concept.active
	• 0 (*Inactive*);
	• 1 (*active*).
	Other aspects of *status* represented by the | Concept inactivation indicator attribute value reference set (foundation metadata concept) |.
	• This set follows the | Attribute value type reference set (foundation metadata concept) |.

<not supported> *RF1* does not support identification of separate modules.	*Concept*. *moduleId*
Concepts.FullySpecifiedName	*Concept*. *Description.term* • Where *Description*.typeid = | *Fully specified name*|; • Configured by *Language Reference Set* • The original FullySpecifiedName (which forms the point of reference for the meaning of the *concept* is the FullySpecification with the earliest *effectiveTime*.

RF1 Concepts Table	***RF2 Concept File***
Concepts.Ctv3Id	| CTV3 simple map reference set (foundation metadata concept) | • This set follows the | Simple map type reference set (foundation metadata concept) |
Concepts.SnomedId	| SNOMED RT identifier simple map (foundation metadata concept) | • This set follows the | Simple map type reference set (foundation metadata concept) |
Concepts.*IsPrimitive* • 0 (*fully defined*); • 1 (*Primitive*).	*Concept*.definitionStateId • |Defined|; • |*Primitive*|.

Table 214: Map between RF1 Descriptions Table and RF2 Description file

RF1 Descriptions Table	***RF2 Description File***
Descriptions. *DescriptionId*	*Description*.id
<not supported> *RF1* releases contain a snap-shot view of the state of each *description* at the time of release.	*Description*. *effectiveTime*

Descriptions. *DescriptionStatus*
• *active*:
  • 0 (*Current*);
  • 11 (*Pending Move*).
• *Inactive*:
  • All other values.
	Description.*active*
	• 0 (*Inactive*);
	• 1 (*active*).
	Other aspects of *status* represented by the | Description inactivation indicator reference set |
	• This set follows the | Attribute Value (reference set pattern) |

<not supported> *RF1* does not support identification of separate modules.	*Description*.*moduleId*
Descriptions.*ConceptId*	*Description*.conceptId
Descriptions.*Term*	*Description*.*term*

<table>
<tr><th>RF1 Descriptions Table</th><th>RF2 Description File</th></tr>
<tr><td>Descriptions. InitialCapitalStatus
• 0 (Initial character case insensitive);
• 1 (Case sensitive);
• <other values not supported>.</td><td>Descriptions. caseSignificanceId
• | Initial character case insensitive |
• | Case sensitive |
• | Case insensitive |</td></tr>
<tr><td>Descriptions. DescriptionType
• Synonym (based on RF2 naming):
  • 0 (Not used in language /dialect):
    • RF2 - Omitted from language /dialect Refset
  • 1 (Preferred term in language /dialect):
    • RF2 - |Preferred| in language /dialect Refset
  • 2 (Synonym in language /dialect) :
    • RF2 - |Acceptable| in language /dialect Refset
• 3 (Fully specified name).</td><td>Description.typeId
• | Fully specified name |
• | Synonym |
• | Definition |
Acceptability in language /dialect represented in | Language type reference set (foundation metadata concept) | for the specified language and dialect.</td></tr>
<tr><td>Descriptions. LanguageCode
• Includes dialect where relevant;
• Language Subsets recommended for representing preferences in dialects.</td><td>Description.languageCode
• Language only not including dialect
• Dialect represented by a | Language type reference set (foundation metadata concept) | for the specified language and dialect.</td></tr>
</table>

Table 215: Map between RF1 relationships tables and RF2 Relationship file

<table>
<tr><th>RF1 Relationships Table</th><th>RF2 Relationship File</th></tr>
<tr><td>Relationships. RelationshipId</td><td>Relationship.id</td></tr>
<tr><td><not supported>
RF1 releases contain a snap-shot view of the state of each active relationship at the time of release.</td><td>Relationship. effectiveTime</td></tr>
</table>

RF1 Relationships Table	***RF2 Relationship File***
<not supported> Only *active Relationship* rows are included in the distribution file for each version.	*Relationship.active* • 0 (*Inactive*); • 1 (*active*). Other aspects of *status* may in future be represented by the | Relationship inactivation indicator attribute value reference set (foundation metadata concept) | • This set follows the | Attribute value type reference set (foundation metadata concept) |
<not supported> *RF1* does not support identification of separate modules.	*Relationship.moduleId*
Relationships.ConceptId1	*Relationship.sourceId*
Relationships.RelationshipType	*Relationship.typeId*
Relationships.ConceptId2	*Relationship. destinationId*

Relationships. CharacteristicType
• 0 (Defining):
  • Inferred - in *Relationships Table*
  • Stated - in separate Stated *Relationships Table*
• 1 (Qualifying).
• 2 (Historical).
• 3 (Additional).
	Relationship. characteristicTypeId
	• |Inferred *relationship*|;
	• |Stated *relationship*|;
	• |Qualifying *relationship*|;
	• |Additional *relationship*|.

Relationships. Refinability	Represented by the | Relationship refinability attribute value reference set (foundation metadata concept) |.
<not supported>	*modifierId*

3.3.4.2 Addition of effectiveTime and active fields

The *effectiveTime* and *active* fields enable the use of a "log style" append-only data model to track all changes to each component for full traceability. Historic data will be supplied in the *RF2 release files*, dating back to the first release in 2002.

Once released, a row in any of the *RF2 release files* will remain unchanged through future releases. In order to change certain properties of a current component (and, therefore, to create a new version of it), a new row must be added to the applicable file, containing the updated data. The *active* field in the newly added row is set to true and the timestamp in the effectiveTimefield indicates the point in time at which the new version comes into effect.

By contrast, 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 by adding a new row 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 release date at which the final version ceased to be valid.

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

3.3.4.3 Active field

As mentioned above, each file contains a *Boolean active* field, used to indicate whether, after the point in time specified in the effectiveTimefield, the version of the component expressed in the row is *active* or *inactive*.

This field replaces the *status* field with a simple binary state. In the previous *Release Format*, this field was overloaded to enumerate both whether the *concept* was *active*, why it was inactivated, and whether it was about to change (or had changed) authority.

The additional information encoded in *RF1*'s *status* enumeration is represented in *RF2* using the following *reference sets*:

- *Concept* inactivation indicator;
- *Description* inactivation indicator;
- *Relationship* inactivation indicator.

These three *reference sets* conform to the *Attribute Value reference set* pattern, and are further described in the "*SNOMED CT Release Format 2 - Reference Set* Specifications" document.

3.3.4.4 History tables

History tracking in *RF2*'s main files uses a log-style, append-only data model. Therefore, the separate ComponentHistory file that formed part of the original *Release Format* is no longer required with *RF2*.

The associations between *inactive* and *active Concepts* that are currently supported by *Historical Relationship* types (e.g. |SAME AS |, "REPLACED BY) will continue to be supported. References held in the *References table* from an *inactive component* to other equivalent or related *components* that were current in the *Release Version* in which that *component* was inactivated will also continue to be supported.However, both of these associations have now moved from the *Relationships* file and the References file to one of the following |Historical association| *reference sets*.

Table 216: RF1 to RF2 History Field Mappings

***RF1* source**	***RF2* |Historical association| *reference set***
MAYBE A (in *Relationships table*)	|POSSIBLY EQUIVALENT TO association *reference set* |
Refers to ('7' in *References table*)	|REFERS TO *concept* association *reference set* |
Similar to ('3' in *References table*)	|SIMILAR TO association *reference set* |
MOVED FROM (in *Relationships table*) Moved from ('6' in *References table*)	|MOVED FROM association *reference set* |

***RF1* source**	***RF2* |Historical association| *reference set***
MOVED TO (in *Relationships table*) Moved to ('5' in *References table*)	|MOVED TO association *reference set* |
Alternative ('4' in *References table*)	|ALTERNATIVE association *reference set* |
WAS A (in *Relationships table*)	|WAS A association *reference set* |
REPLACED BY (in *Relationships table*); and Replaced by ('1' in *References table*)	|REPLACED BY association *reference set* |
SAME AS (in *Relationships table*) Duplicated by ('2' in *References table*)	|SAME AS association *reference set* |

These *reference sets* all conform to the Association *reference set* pattern, and are further described in the "*SNOMED CT Release Format 2 - Reference Set* Specifications" document.

3.3.4.5 Field naming

Lower camel case has been used for field names in distribution file headers and in documentation that describes these files. File names will use upper camel case (starting with a capital letter). File names have also been altered to use a singular not plural form.

An example of upper Camel Case is ThisIsUpperCamelCase. An example of Lower Camel Case is thisIsLowerCamelCase.

3.3.4.6 Field Ordering

Records in the *Concept*, *Description*, *Relationship* and *Reference Set* member files each start with the following four fields:

- id;
- *effectiveTime*
- *active*
- *moduleId*

The four fields have the following meanings:

- The id field provides a unique *Identifier* for the component described by the record;
- The effectiveTimegives the nominal release date at which this version of the component came into effect;
- The *active* flag states whether the components *active* or *inactive*;
- The *moduleId* identifies the source module in which the component is maintained.

The *Identifier file* does not follow the same format, as it works in a slightly different way to the other files, and is described in more detail in the "*SNOMED CT Release Format 2* - Data Structures Specification" document.

3.3.4.7 Concept enumerations

Concept enumerations have been used across *RF2* to replace *integer* enumerations that can only be understood by referencing external documentation. For example, in *RF1*, a *concept status* value of '4' indicates *concepts* that are *inactive* because they are ambiguous. In *RF2*, *concept* enumeration simply uses *concepts* in a metadata *hierarchy* to represent the enumerated *value set* rather than using arbitrary *integer* values directly. Using *concepts* to represent the enumerated values has the following advantages:

- The terminology is self contained, removing the requirement for external documents to explain the meaning of enumerated values;
- Full *language* handling capabilities are available for the enumerated values' representation, useful for standardized multi-lingual representation, and translation support;
- Machine readable model constructs can be used to further describe and enrich the enumerated values.

The following fields have been converted to *concept* enumerations:

Table 217: RF1 to RF2 enumerated field changes

File	Existing *RF1* field name	*RF2* field name
Concept	*IsPrimitive*	definitionStateId
Description	*DescriptionType*	*typeId*
Description	*InitialCapitalStatus*	*caseSignificanceId*
Relationship	*CharacteristicType*	*characteristicTypeId*

Care should be taken not to confuse *Concept* Enumerations with the *term* "enumeration" as used in representational formats. A *Concept* Enumeration is a *concept* whose immediate *children* represent possible values in a range. Each possible value is represented by a single *child concept*, whose *preferred term* may be used, for example, to enable selection from a pick-list of one or more values from the range.

Mappings from *RF1* values to *RF2 concept* enumerations are given below:

Table 218: RF1 to RF2 enumerated value mappings

RF2 field name	*RF1* value	*RF2* value|
definitionStateId	0	| Defined |
	1	| *Primitive* |
typeId	(no specified value)	| Definition |
	3	| *Fully Specified Name* |
	0, 1 or 2	| *Synonym* |
caseSignificanceId	0	| Initial character case insensitive |
	1	| Case sensitive |
	(no specified value)	| Case insensitive |
charactristicTypeId	3	|Additional *relationship* |
	0	|Inferred *relationship* |

***RF2* field name**	***RF1* value**	***RF2* value|**
	0	|Stated *relationship* |
	1	|Qualifying *relationship* |
	2	(no specified value) - now modeled through the *inactive* association *reference set*.

3.3.4.8 Reference Set Data Structures

3.3.4.8.1 Overview of Reference Sets

Reference Set data structures provide the foundation pieces for *RF2*'s generic extensibility mechanism. These building blocks provide a common foundation for *extension* builders to extend *SNOMED CT*, and provide *RF2* with the capability to grow with the *IHTSDO*'s requirements over time.

Conventions applied to the *RF2* files such as field naming, field ordering and history tracking have also been applied to the *Reference Set* specification. This has been done to provide consistency across all components in the *Release Format*.

Generic data structures for *Reference Sets* have been used to create a simple core structure that can be extended to meet a variety of requirements, rather than a complex and inextensible structure that can only be used in a finite and constrained number of ways to enforce editorial policy. This stems directly from a desire to decrease impact on the *SNOMED CT* community by being able to meet future requirements without having to alter the underlying data structures.

Using these generic structures, it is possible to extend the data stored within the main files of *SNOMED CT* to satisfy new use cases without altering the primary structure itself. Containing this extended information in externalised structures such as *Reference Sets* also enables terminology consumers to opt in or out of the content without burdening the primary files with the content. This prevents users from having to download all content and filter out what they don't want, and instead allows them to import the *extension* content should it be desired.

Reference Sets allow the *SNOMED CT core* data structures to be extended, allowing existing components to be grouped together into a set, each tagged with a number of additional fields. Each of these additional fields may either be another *SNOMED CT component*, a *string* or an *integer*. *Reference set* descriptors are also introduced, providing a way to identify the format and purpose of each additional field in a machine readable way. Examples of *reference set* data structures are provided in the "*SNOMED CT Release Format 2 - Reference Set* Specifications" document.

3.3.4.8.2 RF1 Subsets Representation

The *RF1 Subset* mechanism consists of two tables: a *Subsets table* and a *Subset Members* table. Each row in the *Subsets table* describes a *Subset* and characteristics of that *Subset*, as described in the table below.

Table 219: Subsets Table

Field	***Description***
SubsetId	The unique *SNOMED CT Identifier* for this *Subset*
SubsetOriginalId	The unique *SNOMED CT Identifier* for the original *Subset* of which this *Subset* is a version.

SubsetVersion	An *integer* incremented for each revised release of a *Subset*
SubsetName	A name that describes the purpose or usage of this *Subset*.
SubsetType	Indicates the nature of the *Subset* and the type of *SNOMED CT Component* that may be a member of the *Subset*.
LanguageCode	Identifies the *Language* and optionally the *Dialect* to which the *Subset* applies (only used for *description* -based *subsets*: *Language*, *Realm Description*, and *Realm Concept*).
RealmId	Identifies the *Realm* to which the *Subset* applies.
ContextId	May identify the *Context Domain* to which the *Subset* applies

Each row in the *Subset Members* table sets the *status* of a member of an identified *Subset*.

Table 220: Subset Members Table

Field	***Description***
SubsetId	The unique *SNOMED CT Identifier* for this *Subset*
MemberId	The *SNOMED CT Identifier* of this *Subset Member*. This may be a *Concept Identifier*, *Description Identifier* or *RelationshipId*.
MemberStatus	An *integer* specifying the *status*, type or *order* of this member.
LinkedId	Valid for *Navigation* and *Duplicate Terms Subsets* only. For *Navigation Subsets* it is the *SNOMED CT Identifier* for a *Concept* that is a *Navigation child* of the *Subset Member*. For *Duplicate Terms Subsets* it is the *SNOMED CT Identifier* for the highest priority *Descriptions* haring theDuplicate *Term*.

Some *Subsets* and their members are generated automatically from an XML definition file.

3.3.4.8.3 Representing Subsets as Reference Sets

An existing *RF1 Subset* may be represented as an *RF2 Reference Set* in the following way:

A *concept* should be created in the | *Reference Set* | metadata *hierarchy*, using information in the *Subset* table record. A Descriptor for the *Reference Set* should also be set up using information in the *Subset* table record. Then, one *Reference Set* member record should be created for each *Subset Member* table record.

The way in which the *subsets* are represented in *RF2* depends on the *SubsetType* value, as follows:

Table 221: Representing Subsets as Reference Sets

***SubsetType* value**	***Description***	**Mapping to *RF2***
1	*Language*	*Language* type *Reference Set*

***SubsetType* value**	***Description***	**Mapping to *RF2***
2	*Realm Concept*	Ordered type *Reference Set*
3	*Realm Description*	*Language* type *Reference Set*
4	*Realm Relationship*	Ordered type *Reference Set*
5	Context *Concept*	Ordered type *Reference Set*
6	Context *Description*	*Language* type *Reference Set*
7	*Navigation*	Ordered type *Reference Set*.
8	*Duplicate terms*	Ordered type *Reference Set*

3.3.4.8.4 Representing Subsets as Ordered type Reference Sets

| Ordered type | *Reference Sets* can be set up as follows:

First, set up a new *concept* for the *Reference Set* in the |Ordered type| metadata *hierarchy*. The position in the *hierarchy* should be given by the *RealmId* and ContextIdfields in the *Subset* record, as follows:

SNOMED CT Model *component*

Foundation metadata *concept*

Reference set

Ordered type

RealmId

ContextId

If either the *RealmId* field or the *ContextId* fields are "0", "1", blank or null in the *Subset* record, then that level should not be set up in the metadata *hierarchy*. If a *concept* already exists under |Ordered type| with a matching *RealmId* and *ContextId*, then the new *Reference Set* should be set up in that position (as opposed to creating two |Ordered type| *children* with duplicate names).

First, the *concept* describing the *Reference Set* should be created with the following values:

Table 222: Reference Set Concept

Field	**Data type**	**Set to**
id	*SCTID*	A unique id in your *namespace*.

Field	Data type	Set to
effectiveTime	*Time*	The nominal date of release for your *Reference Set*. If a full state valid representation of a *subset*'s history is required, then each previous release of the *Subset* files must be processed in turn (by identifying *Subset* records with a matching *SubsetOriginalId*, in their SubsetVersion *order*), and each amended version must be applied to the *reference set* by appending rows in the usual fashion. The effectiveTimeof each applied change should be set to the date that each version of the *Subset* was released.
active	*Boolean*	'1'
moduleId	*SCTID*	The module *Identifier* for your authoring organization .

Then, add up two *Descriptions* for the FSN and the *Preferred Term* of the *concept*:

Table 223: Reference Set Descriptions

Field	Data type	Set to
id	*SCTID*	A unique id in your *namespace*.
effectiveTime	*Time*	The nominal date of release for your *Reference Set*.
active	*Boolean*	'1'
moduleId	*SCTID*	The module *Identifier* for your authoring organization .
conceptId	*SCTID*	The *Identifier* of the *concept* describing the *Reference Set* that you've just added.
languageCode	*String*	The *language* of the *Description*. This field should be set to the *language* that the *Subset* was defined in, for example - 'en' for English.
typeId	*SCTID*	Create two *Description* records, one for each of the following types: | Fully specified name |, | *Synonym* |.
term	*String*	The *term* for the | *Synonym* | should be set to the *SubsetName* field in the *Subset* record. The *term* for the |FSN| should be set to the same, but appended with "*reference set* (foundation metadata *concept*)".

Finally, add one *Reference Set* member record for each record in the *Subset Members* table for the *Subset*:

Table 224: Converting a Priority Subset to an Ordered Reference Set

Field	Data type	How to populate
id	*UUID*	A new unique *Identifier*
effectiveTime	*Time*	The nominal date on which this release was made.
active	*Boolean*	'1'
moduleId	*SCTID*	Set to the *moduleId* of the authoring organization .
refsetId	*SCTID*	A reference to the *concept* describing the *Reference Set* that you've just created.
referencedComponentId	*SCTID*	Set to *MemberId* in the *Subset Members* table record.
order	*Integer*	Set to *MemberStatus* in the *Subset Members* table record.

Note: Although a *Navigation Subset* can be represented in an |Ordered type| *reference set* as described above, the values of the *linkedTo* field would then have a different meaning, referencing a *child concept* instead of grouping *components* together.

A Descriptor can also be set up for the *Reference Set* if required, as follows:

Table 225: -

refsetId	*referencedComponentId*	*attributeDescription*	*attributeType*	*attributeOrder*
| *Reference set* descriptor|	*Concept describing refset*	| Referenced component |	| *component* type|	0
| *Reference set* descriptor|	*Concept describing refset*	| *Order* |	|Unsigned *integer* |	1
| *Reference set* descriptor|	*Concept describing refset*	| Linked to |	| *component* type|	2

Where *Concept describing refset* is the *Concept* that you've just set up to describe the *Reference Set*. The | *Order* | and |Linked to| *concepts* that describe each additional *Attribute* in the *Reference Set* can also be replaced by more descriptive ones if required. To do this, create the new *concepts* describing the additional fields under the | *Reference set Attribute* | metadata *hierarchy*.

3.3.4.8.5 Representing Subsets as Language type Reference Sets

Language type *Reference Sets* can be set up in a similar fashion to the above, with the following exceptions:

The *LanguageCode* field in the *Subset* record should be used to link the *Reference Set*'s *concept* into the appropriate place in the | *Language* type| metadata *sub-hierarchy*. For example, a value of "en-US" in the *LanguageCode* field would result in the *Reference Set*'s *concept* being created under |US English|:

SNOMED CT Model *component*

Foundation metadata *concept*

Reference Set

Language type

English

US English

RealmId

ContextId

- Where the *SubsetType* is " *Language*" and the *LanguageCode* is a single level (e.g. "en"), then the *Reference Set* should be created at the first level, under |English| in the example above;
- Where the *SubsetType* is " *Language*" and the *LanguageCode* is a two level (e.g. "en-US"), then the *Reference Set* should be created at the second level, under |US English| in the example above;
- Where the *SubsetType* is "*Realm Description* ", then the *Reference Set* should be created under *RealmId* in the example above (where *RealmId* is the value of the RealmIdfield in the *Subset* record);
- Where the *SubsetType* is "Context *Description* ", then the *Reference Set* should be created under *ContextId* in the example above (where *ContextId* is the value of the *ContextId* field in the *Subset* record and *RealmId* is the value of the RealmIdfield in the *Subset* record).

The *Reference Set* member records should be created as described in the following table:

Table 226: Converting a Language Subset to a Language Reference Set

Field	Data type	How to populate
id	*UUID*	A new unique *Identifier*
effectiveTime	*Time*	The nominal date on which this release was made.
active	*Boolean*	'1'
moduleId	*SCTID*	Set to the *moduleId* of the authoring organization .
refsetId	*SCTID*	A reference to the *concept* describing the *Reference Set* that you've just created.
referencedComponentId	*SCTID*	Set to *MemberId* in the *Subset Members* table record.

A Descriptor can also be set up if required.

3.3.4.9 Metadata hierarchy

As the *RF2* data structures and extensibility mechanism contain a number of *concept* enumerations, it is necessary to define *concepts* that represent these values. As well as the enumerated values, there are other machine-readable *concept model* structures not visible in the *Release Format* that require metadata (for example, the structures that define the format of a *description* type).

To meet this need, a new top-level *hierarchy* has been defined as a sibling to the | *SNOMED CT Concept* |, called | *SNOMED CT* Model *component* |. Note that existing metadata *concepts* held within the | *SNOMED CT Concept* | *sub-hierarchy* (|Linkage| and | *Namespace* |) will be moved to the | *SNOMED CT* Model *component* | *sub-hierarchy*.

The top level of the *SNOMED CT* Model component *hierarchy* is structured as follows:

- 138875005 | SNOMED CT Concept (SNOMED RT+CTV3) |
  - 900000000000441003 | SNOMED CT Model Component (metadata) |
    - 900000000000442005 | Core metadata concept (core metadata concept) | ...
      - (*Concept enumerations* required to support *SNOMED CT International Release* data structures)
    - 900000000000454005 | Foundation metadata concept (foundation metadata concept) | ...
      - (metadata required by the *Reference Set* extensibility mechanism)
    - 106237007 | Linkage concept (linkage concept) | ...
      - 246061005 | Attribute (attribute) | ...
      - 416698001 | Link assertion (link assertion) | ...
    - 370136006 | Namespace concept (namespace concept) | ...

Figure 43: SNOMED CT Model Component Hierarchy

Note that only | is a | *relationships* will exist between *concepts* in the | SNOMED CT Model Component || *hierarchy* |. Other associations between *concepts* in this *hierarchy* can be modeled using an | Association type reference set (foundation metadata concept) **|** (see *Association Reference Set*).

3.3.4.10 SCTIDs and UUIDs

UUIDs are unique universal *Identifiers*. These 128 bit unsigned *integers* can be used to identify all *SNOMED CT components* internally.

SCTIDs will continue to be used as primary and foreign keys for *concepts* and *descriptions*, both to identify a component and to reference other components. This form is essential for vendors and implementers who will reference *concepts* in *Clinical Information Systems* and messages. *SCTIDs* will also be used to identify *relationships*. However, *UUIDs* will be used to identify *Reference Set* members.

In addition, any *UUIDs* used in development can also be published as additional *Identifiers* via the *Identifier file*.

3.3.4.11 Description text

The values permitted within the *description term* field have been extended to support arbitrary length content, and support mark-up content such as XHTML. The 900000000000538005 | Description format reference set | allows a maximum length and format to be associated with each *description* type within the *Description file* (see *Description format reference set specification*) .

This mechanism allows descriptive text of different formats (other than *Fully Specified Names* and *Synonyms*) to be associated with *concepts*, while appropriately constraining existing *description* types. This enables all *descriptions* associated with *concepts* that may require translation to be held in one place in the *Description file*.

3.3.4.12 LanguageCode

The languageCode field is retained in the *Description file*, but is restricted to contain only coarse-grained *language* information (e.g. "English" or "French"). *Reference sets* are used to indicate *dialects* and contexts, where required. As an example, the *term* "Bulldozer" would appear once in the *Descriptions* file with the *language* code en ("English"), but be listed separately in each of the Australian, UK and US English *language* national *dialect Reference Sets* as a valid *term* in all three *dialects*.

The languageCode field in *RF2* is a text field and is bound to the *ISO* 639-1 two-character *language* codes.

3.3.4.13 Addition of a modifierId field

The underlying semantics on which *SNOMED CT* is based assumes that all *relationships* are existential restrictions. In other words, a *relationship* in *SNOMED CT* implies that there is ***some*** instance of that *relationship* from each instance of the source *concept* to any instance of the target *concept*. Other types of *relationship*, such as universal restrictions do exist and have been studied extensively. For example, the existence of a universal *relationship* in *SNOMED CT* would require that ***all*** instances of that *relationship* from each instance of the source *concept* be to an instance of the target *concept*.

As an example, take the following hypothetical *relationship* |Has *child* | between two *concepts* | Woman | and | Girl |:

| Woman | | Has child | = | Girl |

In *SNOMED CT*, the *relationship* is implicitly an existential *relationship*, that we can make explicit in the above syntax by adding the modifier "some:", as follows:

| Woman | **some:** | Has child | = | Girl |

This means that every instance of | Woman | has at least one | Has child | *relationship* that has as its target an instance of | Girl |. In other words, in our hypothetical world, every woman would have at least one daughter, but may also have any number of sons.

If the existential *relationship* were changed to a universal *relationship*, as follows, then the meaning would be changed:

| Woman | **all:** | Has child | = | Girl |

This means that, for every instance of | Woman |, all its | Has child | *relationships* must have a target of | Girl | . In other words, in our hypothetical world, women could only have daughters or no *children* at all, and could not have sons. This has a very different meaning from the existential *relationship* currently implied within *SNOMED CT*.

A new *modifierId* field has been added to the *Relationship file* to allow future expansion. This *concept* enumeration field will initially be set to | Some | to keep compatibility with the existing semantics of *SNOMED CT*. Widening the range of this field to include other values (such as | All |) would 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. Therefore, before extending the range of this field beyond | Some |, a test of the impact on tooling will need to be performed, and the results reviewed and approved.

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 | Some | will be discussed with *Members* first and approved by the Technical Committee.

3.3.4.14 Addition of moduleId field

A *moduleId* field has been added to help identify content and dependencies in a release. This enables release centers to compose a unified release (in a single set of *release files*) from a number of different modules, yet still identify the origin of content down to a row level within each of the releases. For example, this may be used to differentiate *SNOMED CT* International content, Australian Medicines terminology and Pathology content within the Australian *National release*. Currently this is only possible if all modules are assigned unique sub - *namespaces*, and content consumers parse *Identifier namespaces* to differentiate modules.

Components may move from one module to another within a particular *namespace*. Without a *moduleId*, there would be a need to retire a component in one *namespace*, and add another (with a new *SCTID*) to the *namespace* that the component is moved to. Additional *relationships* would also need to be set up, to link the old and new components together. None of this administrative and error-prone work is required if *moduleIds* are used.

Combining the *moduleId* with *Reference Sets* provides a powerful versioning mechanism. The Module Dependency *reference set* (described in more detail in the *SNOMED CT Release Format 2* - Reference SetSpecifications" document) can represent interdependencies between modules and define compatible versions. This functionality can thus be used to represent version information for a terminology's components within the terminology's content itself, in a machine processable way.

The diagram below provides an example of this structure. It shows the components making up an Australian national *SNOMED CT extension* release, containing subcomponents. The links can be described using members of the Module Dependency *Reference Set*. In the example below:

- *SNOMED CT* Australian *Extension* depends upon *SNOMED CT* International 2008-01-31;
- Australian Pathology *Extension* depends upon *SNOMED CT* Australian *Extension* 2008-08-31;
- Australian Discharge Summary *Extension* depends upon *SNOMED CT* Australian *Extension* 2008-08-31.

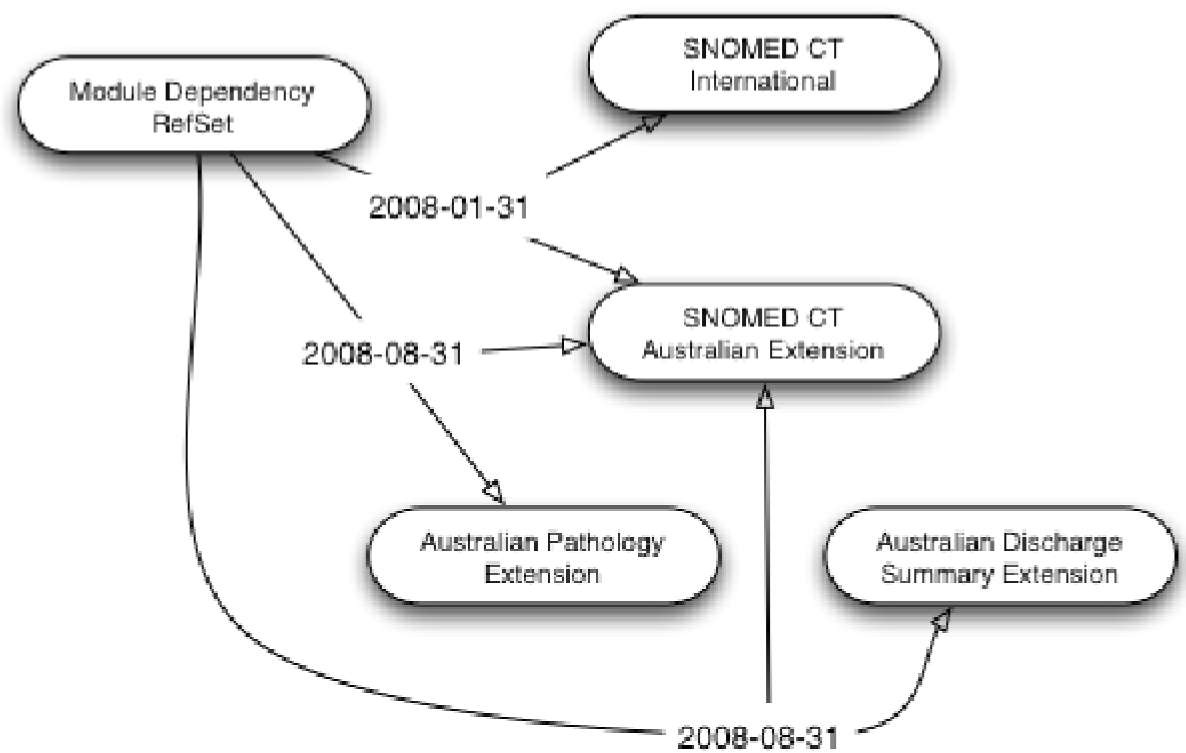


Figure 44: Illustration of module dependencies

3.3.4.15 Fully Specified Names and Preferred Terms

RF2, like the original *Release Format*, allows *Fully Specified Names* (FSNs) to be specified in each *language* using the *Description file*. Multiple FSNs and multiple *synonyms* may exist with the same languageCode for a *concept*. However, a particular *language Reference Set* will only contain a single FSN and a single *preferred term* for a *concept*.

As part of the *language* modifications made in the *RF2*, only a broad definition of a *language* can be made for a *Description*. For example, it is possible to declare a *Description* as English, but not US English. Also *RF2* no longer contains a *description* type value for a "*Preferred Term* ", only types of | *Fully Specified Name*

| and | *Synonym* |. Each *Synonym* can then be assigned an |Acceptability value| of either |Acceptable| or |Preferred| when included in a *language reference set*.

As a result of these changes, the preference for particular *descriptions* in a *language* or *dialect* is now represented using a *Reference Set*. This matches the specified use of *Language Subsets* in *RF1*, and deliberately removes the deprecated approach applied in some releases where preferences were derived directly from the released *Descriptions* file.

Language reference sets also introduce the notion of overriding or inactivating particular *Descriptions* that may be appropriate in one *dialect*, but not appropriate in another dependent *dialect* or context. This is achieved by allowing *Descriptions* that are inherited from a parent *language reference set* to be overridden in a *child language reference set*.

3.3.4.16 Field removals

A number of fields that appeared in the previous *Release Format* do not appear in *RF2*. These fields are listed in the table below, with an explanation of why each field has been removed and to where it has been moved. Note that where a *reference set* replaces a field, this *reference set* will be provided with the *RF2* distribution.

Table 227: RF1 fields that are not use in RF2

File	Field	Rationale for change	Moved to
Concept	CTV3Id	To avoid cluttering the *concept* table.	Moved to the | *CTV3* simple map| *Reference Set*.
Concept	*SNOMEDID*	To avoid cluttering the *concept* table.	Moved to the | *SNOMED RT* IDsimple map| *Reference Set*.
Concept	FullySpecifiedName	This field duplicates one of the *fully specified names* represented in the *Description file*. This duplication has led to misunderstanding of the use of *fully specified names* in multi-lingual distributions of *SNOMED CT*.	The original FSN, which may be required for translation purposes, can be identified as the FSN for the *concept* that has the earliest *effectiveTime*.
Relationship	*Refinability*	As this information is only useful in some environments, it has been moved out of the *Relationship file* to avoid cluttering it.	Moved to the | *Relationship refinability reference set* |.

3.3.4.17 Identifier file

The *Identifier file* has been added to provide a standardized means of attaching co-referent *Identifiers* from many different schemes to *SNOMED CT components*. This provides a means to:

- link *UUIDs* and *SCTIDs*, and;
- add external *Identifiers* such as *LOINC* codes, where these are truly co-referent; and;
- track history and organizational responsibility by linking old *SCTIDs* to new ones, where components are transferred from one name space to another, in order to allow uninterrupted use of the old *SCTIDs*.

This provides a mechanism for generically binding *SNOMED CT components* to an arbitrary number of alternative *Identifiers*. It is a more scalable solution than appending columns as needed to the *Concept file*.

Note that the *Identifier file* is not intended as a mapping solution. This structure is only intended to support cases where the external *Identifier* means exactly the same thing as the *SNOMED CT* component to which it is attached. For example, it is not envisioned that ICD-9, *ICD-10* or *CTV3* codes would be entered into this file.

The *Identifier file* is intended to provide a mechanism to represent external codes for *SNOMED CT components* where the meaning is exactly the same. For example, in the Australian Medicines terminology (AMT), *concepts* are "generated" from data sourced from the Therapeutic Goods Administration (TGA) and the TGA has an ARTGID for every therapeutic item. This mechanism allows the ARTGIDs to be attached directly to the corresponding AMT *concept* when generated. In this instance, the *Identifier file* assists meeting the use case without burdening the *descriptions* file or *concepts* file with this content.

3.3.4.18 References table

In the previous *Release Format*, the *References Table* contained References from *inactive components* to other equivalent or related *components* that were current in the *Release Version* in which that *component* was inactivated. Each Reference indicated the nature of the *relationship* between the *inactive* and persistent component.

In *RF2*, this information is held in *Historical Association Reference Sets*.

3.3.4.19 Textual Descriptions

In the previous *Release Format*, a separate Textual *Descriptions* file held long *descriptions* (of up to 512 bytes, in plain text format). In *RF2*, these textual *descriptions* are transferred to the *Description file*.

3.3.4.20 Mapping

3.3.4.20.1 Mapping Overview

No bespoke mapping file structures (for example, CrossMapSets tables) have been defined in *RF2*. Instead, the simple map *Reference Set* pattern and alternate map *Reference Set* pattern should be used, in conjunction with other *Reference Set* patterns, to define *Reference Sets* for mapping purposes. See the "*SNOMED CT Release Format 2 - Reference Set* Specifications" document for more details.

3.3.4.20.2 Mapping in RF2

RF1 CrossMaps that have a type of either one-to-one or one-to-many can be represented in *RF2* as described below. The type of an *RF1 CrossMap* can be identified from the *MapSetType* field in the CrossMapSets table. The following values in the *MapSetType* field are possible:

Table 228: RF2 Mapping Type Representations

Value	Meaning	Examples	Mapped to *RF2*
1	One-to-one	ICD-O	Can be mapped automatically, as described below
2	One-to-many	ICD-9-CM	Can be mapped automatically, as described below
3	Alternate on-to-one maps	None known of	Can be mapped automatically. Further definition will be given if necessary.
4	Alternate one-to-many	None known of	May need manual intervention to map.

For *CrossMaps* that have a *MapSetType* of either '1' or '2', first, create a *concept* under the **|**Complex map**|** *sub-hierarchy* to describe the Complex map *Reference Set*, in the following location:

SNOMED CT Model *component*

Foundation metadata *concept*

Reference Set

Complex map

MapSetRealmId

Where *MapSetRealmId* is set to the contents of the *MapSetRealmId* field in the CrossMapsSets record of the *CrossMap* to be represented in *RF2* format. Where the *MapSetRealmId* field is blank or null, then an intermediate *concept* should not be created, and the *Map Reference Set concept* should be created as a direct *child* of **|**Complex map**|**. The *concept* should be created as follows:

Table 229: RF2 Map Versioning

Field	Data type	Set to
id	*SCTID*	A unique id in your *namespace*.
effectiveTime	*Time*	The nominal date of release for your *Map Reference set*. The year of the nominal release should tie up with the year in the MapSetSchemeVersionfield in the *CrossMapSets* record.
active	*Boolean*	'1'
moduleId	*SCTID*	The module *Identifier* for your authoring organization .

Once the *concept* is created, add two *Descriptions* for the FSN and a *Synonym*.

Table 230: RF2 Mapping Metadata

Field	Data type	Set to
id	*SCTID*	A unique id in your *namespace*.
effectiveTime	*Time*	The nominal date of release for your *reference set*.
active	*Boolean*	'1'
moduleId	*SCTID*	The module *Identifier* for your authoring organization .
conceptId	*SCTID*	The *Identifier* of the *concept* describing the *Reference Set* that you've just added.
languageCode	*String*	The *language* of the *Description*.

Field	Data type	Set to
typeId	*SCTID*	Create two *descriptions*, with each of the following types: |FSN|, | *Synonym* |.
term	*String*	*Terms* for the FSN and the *Synonym*. The *Synonym* should be set to the *MapSetName* in the *CrossMapSets* record. The FSN should be set to: MapSetSchemeName + "("+ *MapSetSchemeId* + ")" + " *reference set* (foundation metadata *concept*)".

Finally, add members to the *Reference Set* that you've just created.

To do this, identify each CrossMaps table record with a *MapSetId* that matches the *MapSetId* field in the CrossMapsSets record for the *CrossMap* that you're representing in *RF2*. For each CrossMap table record, identify the related CrossMapTarget record using the *MapTargetId* field in the CrossMaps record. The *TargetCodes* field in the CrossMapTarget record will contain zero or more *target codes*, each separated by a separator character identified by the *MapSetSeparator* field of the CrossMapSets record.

One *Reference Set* member record should be created for each *target code* identified within the *TargetCodes* field, as follows:

Table 231: RF2 Mapping Representation

Field	Data type	Purpose
id	*UUID*	A unique *UUID* for the new member record.
effectiveTime	*Time*	The nominal date of release that this member is to be first introduced in.
active	*Boolean*	'1'
moduleId	*SCTID*	The module *Identifier* for your authoring organization .
refsetId	*SCTID*	The id of the *concept* that describes the *Reference Set* that you've just created.
referencedComponentId	*SCTID*	Set to the *MapConceptId* in the CrossMaps record.
mapGroup	*Integer*	This field should be set to '1' for the first *target code* within *TargetCodes* field of the CrossMapTargets record. If there is more than one *target code* in the field (separated by a separator character), then this field should be set to '2', '3', etc. For each subsequent code.
mapPriority	*Integer*	'1'
mapRule	*String*	Set to null

Field	Data type	Purpose
mapAdvice	*String*	Set to null
mapTarget	*String*	Set to the *target code* in the *TargetCodes* field of the CrossMapTargets record.

3.3.4.21 Release Types

Release Format 1 only supports a single *Release Type* which represented the entire set of currently relevant components. In contrast *Release Format 2* supports three different *Release Types* including a full historical view of all components ever released and a *delta release* that contains only the changes from one release to another.

The *Release Format 2 Specification* describes the *Release types* and the *Terminology Services Guide* provides advice on *importing different Release types*.

3.3.4.22 Interchange format

RF2 is conceived as a replacement for the current *Release Format*. It is designed to provide a way to publish releases of *SNOMED CT Release* to implementers and other licensees. There is a close *relationship* between the requirements to support distribution of content and the requirements for exchanging *components* during content development. However, there are also significant differences related to the requirement for additional development information (author, change time, etc) and a need to support work with 'interim' incomplete and unpublished *components* which have not yet been assigned a *SNOMED CT identifier*.

Previous *IHTSDO* work resulted in a draft specification of *SNOMED* Interchange Format (SIF) which addressed some of these issues. Some of the provisions of *RF2* are already supported by SIF but others will require revisions to the SIF specification.

3.3.4.23 Post Coordinated expression Syntax

RF2 allows *relationship types* to be extended from "existential qualification" to other types of *relationship* such as "universal qualification". This *extension* will not be used in initial releases until the complexity of the underlying semantics has been fully tested, but once it is introduced, post coordinated *expression* syntax will also need to be extended to cater for this.

3.3.5 RF1 Compatibility and Conversion Tools

In January 2012 the *IHTSDO* 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 *IHTSDO* recognizes that, while implementers will wish to benefit from the features of the new format, there is inevitably a transitional period during which both format are in use. Therefore, the *IHTSDO* 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 *Identifiers* of RF1 derivatives (Subsets

and *Cross Maps*) differ from those used for the equivalent *Reference Sets* in RF2. These differences are an essential consequence of ensuring that the RF1 data produced by conversion from RF2 is fully compatibility with existing RF1 systems.

- The RF2 to RF1 Conversion Tool used for generating the RF1 files is also available to all *IHTSDO 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 *IHTSDO 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.

3.3.5.1 Relationship Refinability Reference Set

The *Relationship Refinability Reference Set* is included in the RF1 Compatibility Pack. It provides information about whether it is permissible to refine the value of a *Relationship*. This information is equivalent to the *Release Format 1 Relationships*. *refinability* field. It is not included in the main RF2 release as it value is likely to diminish over time as the *Machine Readable Concept Model* provides a more complete representation of *refinability*.

This *Reference Set* is identified as 900000000000488004 | Relationship refinability attribute value reference set (foundation metadata concept) | and its *Concept enumeration* values are specified in *Table 232*.

Table 232: Refinability value (foundation metadata concept) (900000000000226000)

Id	Term	Comment
900000000000007000	| Not refinable (foundation metadata concept) |	The value provided by the *destinationId* may be used but none of the *subtypes* of this *concept* are permitted.
900000000000218008	| Mandatory refinability (foundation metadata concept) |	The value may be refined by selecting a *subtype* of the *concept* referred to by the *destinationId*.
900000000000216007	| Optional refinability (foundation metadata concept) |	The value may be refined by selecting a *subtype* of the *concept* referred to by the *destinationId*.

3.4 SNOMED CT - File Naming Conventions

The file naming convention specified in this section applies to all *IHTSDO release files* starting with the January 2010 *International release*.

The specification 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* by which a *release file* is owned.
- A consistent versioning mechanism.
- An easy 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 names will be unique across the *International release* and releases from individual *National release* centers and across separate releases from each center over time.
- An upgrade path, to enable use of the same naming convention with the new *release format* (*RF2*), while enabling easy identification of whether a file is in *RF1* or *RF2* format, and avoiding naming clashes.

Quality Assurance checks, to ensure that this naming convention is enforced, will be performed as part of the *International release* process. It is expected that equivalent checks will be performed as part of each *National Release* Center 's release process.

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

3.4.1 File Naming Convention - Overview

3.4.1.1 General File 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>_<Country|Namespace>_<VersionDate>.<Extension>

Each element in the above structure is described in more detail in subsequent sections.

3.4.1.2 General Naming Rules

The following rules apply generally to all elements of the file name:

- All elements are mandatory and may not have a null value;
- Elements of the file name may only contain alphanumeric characters, with the exception of hyphens (" **-**") used in connection with *language* codes (see detail for the ContentSubType element below);
- All text should be in US English, except as explicitly allowed below;
- Abbreviations should not be used, except for specified codes or tags;
- The maximum length of a file name (including separators and *extension*) is 128 characters.

3.4.1.3 Rules for "Readme" Files

Readme files distributed as part of a *SNOMED CT release* have their own specific naming convention, as shown below:

Language is the *language* code for the *language* of the Readme file, as specified below for the ContentSubType element, and the VersionDate corresponds to the version date of the release.

3.4.2 FileType element

3.4.2.1 Description

The FileType element of the filename designates the type and intended use of the *release file*. It consists of a 3-5 letter code and must be lowercase.

3.4.2.2 Rules (Release Format 1)

Allowable FileType codes are shown in the table below:

Table 233: Allowable File Type Codes

Code	File Type *Description*
"sct" + <format tag>	Terminology Data File
"der" + <format tag>	Derivative Work Data File
"res" (+ <format tag>)	Implementation Resource Data File
"tls" (+<format tag>)	Implementation Resource Tool
"doc" (+<format tag>)	Documentation
"z" + Code	Archival/Unsupported File (e.g. zsct)
"x" + Code	Test/Beta Release File (e.g. xder)

The allowable file types are described in more detail below:

- **Terminology Data File**(" **sct**") - the set of *Release Format 1* files that make up the *SNOMED CT* terminology. These are:
  - *Concepts table*
  - *Relationships table*
  - *Descriptions table*
  - *Component History table*
  - *References table*.
- ***Derivative* Work Data File**(" **der**") - data files that make up a *SNOMED CT* "*derivative* work" (a product for use in conjunction with *SNOMED CT* that cannot be effectively used without the terminology - such as *subsets* or *Cross Maps*). Examples of *Release Format 1* files within this group include:
  - Subsets table;
  - SubsetMembers table;
  - CrossMapSets table;
  - CrossMaps table;
  - CrossMapTargets table;
- **Implementation Resource Data File**(" **res**") - data files intended to support developers with the implementation of *SNOMED CT*, but that are not necessarily useful to end-users. Examples from *Release Format 1* include:

- *Canonical table*
- *Stated Relationships table*

- **Implementation Resource Tool**(" **tls**") - software tools or other files that do not contain original *SNOMED CT* content (i.e. that is not also held elsewhere in the release), but can be of use to implementers. If such files cannot comply with this naming convention (for example, if some other standard applies), then those files should be distributed as part of a ZIP file archive that does conform to this file naming convention. Examples from *Release Format 1* include:
  - Index Generator.
- **Documentation**(" **doc**") - documents defining *SNOMED CT* standards, policies and guidelines, as well as documentation for files or products included in a *SNOMED CT release*. Most, but not all, files in this group are released in a PDF format.
- **Archival/Unsupported File**("**zsct**", "**zder**", "**zres**", "**ztls**") - files that are not currently supported or updated, but may be of some use to implementers. These files should only be used with caution and after appropriate review and validation. The letter "**z**" is inserted in front of the usual FileType code for these files (i.e. "**z**" + "**sct**", "**der**", "**res**" or " **tls**"). Examples from *Release Format 1* include:
  - *SNOMED* 3.5 to *SNOMED RT* bridge file;
  - *SNOMED* 2 to *SNOMED RT* bridge file.
- **Test/Beta *Release File***("**xsct**", "**xder**", "**xres**", "**xtls**") - files distributed as part of a test/beta release, or as a "technology preview". These files should only be used for review and evaluation purposes. The letter "**x**" is inserted in front of the usual FileType code for these files (i.e. "**x**" + "**sct**", "**der**", "**res**" or "**tls**").

3.4.2.3 Format Tags

A *Release Format* tag must be appended at the end of the three-letter FileType code if the file named is dependent on a particular *Release Format* specification. The allowable *Release Format* tags are:

- For files that are part of the current *Release Format* (*RF2*), or applicable only to the *RF2 Release Format*, the number "**2** " is appended to the FileType code (e.g. "**sct2**", "**der2**", "**res2**").
- For files that are part of the now obsolete *RF1 Release Format*, or applicable only to that *Release Format*, the number "**1** " is appended to the FileType code (e.g. "**sct1**", "**der1**", "**tls1**").
- If the file is not specific to either *Release Format*, the three-letter FileType code should be used without a *Release Format* tag (e.g. "**res**", "**tls**" or "**doc**").
- The FileType code for all terminology and *Derivative* Work data files ("**sct** " or "**der**") must include a *Release Format* tag ("**1**" or "**2**"). For other file types, the *Release Format* tag is optional.

3.4.3 ContentType element

3.4.3.1 Description

The ContentType element of the filename describes the content and purpose of the file. It consists of 2-48 alphanumeric characters in camel case.

3.4.3.2 Rules (Release Format 1)

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

- **For Data Files**("**sct**", "**der**" or " **res**") - the name of the table contained in the file should be used as the value for the ContentType element. Possible values for the *RF1 Release Format* are:
  - *Concepts*
  - *Descriptions*
  - *Relationships*

  - SubsetMembers;
  - *Subsets*;
  - ComponentHistory;
  - References;
  - CrossMapSets;
  - CrossMaps;
  - CrossMapTargets;
  - TextDefinitions;
  - Canonical;
  - DualKeyIndex;
  - WordKeyIndex;
  - StatedRelationships.

- **For ImplementationResource Tools**("**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;
- **For Documentation**("**doc**") - the title of the document, which may be abridged but should not be abbreviated, should be used as the value for the ContentType element;
- **For Archival & Test/Beta Files**("**z**"+ code or "**x**"+ code) - the value of the ContentType element should be determined according to the rules for a normal file of the same type ("**sct**", "**der**", "**res**" or "**tls**").

3.4.4 ContentSubType element

3.4.4.1 Description

The ContentSubType element of the filename 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 *language* code). Hyphen ("**-**") is a permitted character in conjunction with a *language* code, as described below.

3.4.4.2 Rules

Most *RF1* data files are adequately identified by the information appearing in the ContentType, so that there is no information that is required to appear in the ContentSubType element. In such cases, a placeholder should be used to avoid giving this element a NULL value. The allowable placeholders are:

- "**Core**" **-** for files that are part of the *SNOMED CT International Release*;
- "**National**" - for files that are part of the *National Release* of an *IHTSDO Member* country; and;
- "**Local**" - for files that are released by an *Affiliate Licensee* or other authorized 3 rd party.

Note: These placeholders must not be used in file names for *Release Format 2* files as in these cases the ContentSubType element indicates the release type.

- **Implementation Resource Tool** ("**tls**") - the value of this 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. If appropriate, the element may contain a *status* tag with one of the values described below under Documentation.
- **Documentation** ("**doc**") - the element should contain at least two components: a *status* tag and a *language* code (see above). Additional components may be added to this element if necessary to fully identify the document. Possible values for the document *status* tag are:
  - "**Current**" - indicates that the document is up-to-date and complete for the *current* release of *SNOMED CT*, as indicated by the VersionDate element;
  - "**Draft**" - indicates that the document is a draft version; it may be incomplete and has not been approved in a final version;

- "**Review**" - indicates that the document has been released for review and comments from *SNOMED CT* users and other stakeholders.

- **Archival** & **Test/Beta File** ("**z**"+ code or "**x**"+ code) - the value of the element should be determined according to the rules for a normal file of the same type ("**sct**", "**der**", "**res**" or "**tls**").

3.4.4.3 Language Usage

For files released as part of a National or local release, and which do not appear in the *SNOMED CT International Release*, the value of the ContentSubType element may be given in a *language* other than English, with the following limitation:

- Any of the four sets of defined values for the ContentSubType element that are present in the file name may not be translated, but must appear as specified herein. These are: *language* code, *Release Type* flags ("**Full**", "**Snapshot**", "**Delta**"), placeholders ("**Core**", "**National**", "**Local**"), and *status* tags ("**Current**", "**Draft**", "**Review**").

3.4.4.4 Language Codes

Where it is necessary to specify the *language* of a file, a *language* code must be included in the ContentSubType element. A *language* code is a *string* identifying the *language* and, if appropriate, the *dialect* of a file, and consists of a code and optionally a sub-code. If a sub-code is present it is separated from the code by a hyphen ("**-** ").

The code is the two-character *ISO* 639-1 *language* code. *ISO* 639 is the International Standard for "Codes for the representation of names of *languages*". The sub code is a *string* of upper-case letters that represent the *dialect*. This deliberately mirrors the W3C approach and will either be:

- If the *dialect* is general to an entire country, the two-letter *ISO* 3166 country code is used. *ISO* 3166 is the International Standard for "Codes for the representation of names of countries".
- If *dialects* are used that are less common or not country or *language* linked, the IANA approach is used; this code consists of a *string* of more than two letters. IANA is the Internet Assigned Numbers Authority.

This structure follows Internet conventions. Examples: "**en**" for English, "**es**" for Spanish, " **en-US** " for United States English, " **en-GB** " for British English.

If the ContentSubType includes more than one component (e.g. document *status* and a *language* code), the *language* code must be the last component in theContentSubType element and should be preceded by a hyphen ("**-**") placed before the *language* code.

3.4.5 Country|Namespace element

3.4.5.1 Description

The Country | *Namespace* element of the filename helps to identify the organization responsible for developing and maintaining the file. Its format is 2-10 alphanumeric characters consisting of 0, 2 or 3 upper-case letters followed by 0 or 7 digits.

3.4.5.2 Rules

The following rules apply to the content of this element:

- Letters, if present, are either the *ISO* -3166 2-character country code for an *IHTSDO Member* country or "INT" for files that are part of the *IHTSDO*'s *International Release* of *SNOMED CT*;
- Digits, if present, are a *SNOMED CT Namespace Identifier*.

Valid combinations are:

- 2 characters only - the file is part of a Member *National Release*, but not part of a specific *Namespace* - this combination is not valid for Data Files ("**sct** ","**der**" or "**res**");

- 3 characters only ("**INT**") - the file is part of the *IHTSDO International Release* and belongs to the International *Namespace*;
- 2 characters and 7 digits - the file is part of a Member *National Release* and belongs to the specified *Namespace*;
- 3 characters and 7 digits - the file is an optional part of the *IHTSDO International Release* and belongs to the specified *Namespace*; or;
- 7 digits only - the file has been developed and released by a 3rd party, identified by the specified *Namespace*.

3.4.6 VersionDate element

3.4.6.1 Description

The VersionDate element of the filename 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.

3.4.6.2 Rules

The following rules apply to the content of this element:

- 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.

VersionDate refers to the official, published date of a *SNOMED CT International Release*, or of the *National Release* of an *IHTSDO Member* country, and may not always correspond to the actual date of distribution of any particular release.

3.4.7 Extension element

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.

Chapter 4

4 Terminology services Guide (RF1)

4.1 Representing SNOMED CT in an application

One of the first steps in the design and development of a *terminology server* is to decide how the *SNOMED CT* resources will be stored and accessed. Options include use of a relational database, an object database or a proprietary file structure. The decision will be influenced by the experience of the developers and the technical environment in which development is undertaken. The following sections briefly outline some of the options and identify general issues that should be considered when making this decision.

Those intending to use a third-party *terminology server* are unlikely to be faced directly with the decision of how to represent the information. However, they will be interested in the consequences of these decisions in terms of performance and the overall size of the installed *SNOMED CT* resource. If a *terminology server* uses a particular relational database with an additional licensing or technical support cost this may also affect its suitability.

4.1.1 Direct use of release files in a relational database

The distributed *release files* can be imported directly into a database schema that matches the distribution file specification. This data then provides core resource at the heart of a *terminology server*.

This direct use of distributed files in a relational database has the advantage of allowing simple installation. However, it may not be the most efficient approach in terms of performance or file size. Some *terminology services* require relatively complex queries with multiple joins, and need to be completed in fractions of a second to provide an acceptable *user interface*.

Example: To display the set of *subtype children* of a *concept* with their *preferred terms* in a specified language or *dialect* requires several joins between *concepts*, *Relationships*, *Descriptions* and a language *refset*.

To search for a term matching a supplied pattern in a *concept* that represents a type of procedure also requires multiple joins to link the *Descriptions* with matching terms to the relevant *concept* and test whether it is a *subtype* of the 71388002 | Procedure (procedure) | *concept*.

The performance criteria of searches and joins in very large relational databases vary significantly. Therefore, different optimizations may need to be used to achieve acceptable response times according to the nature of the relational database system.

4.1.2 Alternative relational structure

There is no requirement to use the data structure as distributed. Other structures can be used provided they are able to deliver the range of *Terminology services* required. Options include:

- More normalized representations of the model.

Example: Excluding the *Fully Specified Name* field from the *Release Format 1 Concepts table* increases normalization of the data, since the *Fully Specified Name* is also available in the *Description* table.

- Less normalized representations of the model.

  Example: Replicating the |is a| *Relationships* with the source and target *ConceptId* fields reversed decreases normalization of the data but might have some performance benefit.

- Omission of some of the tables.

  Example: If a *terminology server* is not required to support *Cross Mapping*, the components required for this functionality may be omitted.

- Replacement of some of the supporting tables with proprietary alternatives.

  Example: The word search support tables could be completely replaced by other tables or indices generated by the *terminology server* when loading the distribution files.

4.1.3 Non-relational structures

Although the primary distribution format is relational, this does not require *terminology servers* to utilize a relational database as the primary or only storage format. The requirements for *terminology services* may also be met by representing some or all of the distributed data in other forms including object-oriented databases, Extensible Mark-up Language (XML) and/or proprietary data structures. These structures may be used separately or, in some cases, in combination with a relational database.

A possible conclusion from analysis is that most *terminology services* are *concept*-centered. One structure that reflects this is a hierarchical design in which all the information about each *concept* is bound together as a discrete unit.

Table 234: Representing SNOMED CT in a hierarchical structure

SNOMED CT				
	[1..n] *Concepts*			
		[1..n] *Descriptions*		
			[1..n] Subsets related to the *Description*)	
				Including support for language and term usage subsets.
		[1..n] *Relationships*		
		[1..n] Subsets related to the *concept*)		
			Including support for subsets and crossmaps.	

Such a structure needs support from indices and/or other data access tools to allow searches. However, it has the merit that whenever a *concept* is accessed, all the relevant information is immediately at hand without following relational joins to multiple rows in other tables.

4.2 Importing and updating from Release Files

4.2.1 Importing distribution files

This section is concerned with the general requirements for importing an *International Release* of *SNOMED CT* so that it can be accessed by a *terminology server*. Separate sections address additional issues related to updates of an existing installation and importing *Extensions Releases*.

4.2.1.1 Requirements for importing distribution files

Terminology servers should be able to import *SNOMED CT distribution files* into the internal form used for the *SNOMED CT* resource.

The import process should be automated with user intervention limited to selection of *Subsets* and/or configuration options.

The import process should import:

- The *core files* (*Concepts*, *Descriptions*, and *Relationships*).
- A selected set of *Subsets*:
    - This import may be either from:
        - A selected set of *Subset Definition Files*
        - Selected rows in the *Subsets Table* and *Subset Members Table*
    - Some *Subsets* may be used to filter the import of other distribution files [15]:
        - *Descriptions* may be filtered to limit coverage to a specified *language* or *dialect*;
        - *Concepts* may be filtered to limit coverage to *Concepts* relevant to human medicine by excluding those only relevant to veterinary medicine;
        - *Relationships* associated with any *Concept* that are excluded by *Subset* filtering should also be filtered out.
    - *Language Subset* may alter the *DescriptionType* and/or *LanguageCode* values in the distributed *Descriptions Table*.
- Any other distribution files needed to support the required set of services.

4.2.1.2 Checking during the import process

The import process should check the imported data to confirm that:

- The distribution files imported all parts of the same release.
- All *SCTIDs* have:
    - A *partition identifier* appropriate to the field;
    - A valid *check-digit*.

Other consistency checks may also be applied to ensure the integrity of the data.

[15] See also guidelines on implementation of Subsets. Language Subsets can be used in single language implementation by filtering and applying the DescriptionTypeand LanguageCodesduring the import process. However, multi-lingual installations may require dynamic application of the alternative values applicable to the supported languages and dialects.

4.2.1.3 Pre-processing of distribution files by terminology server suppliers

The import process may be time -consuming due to the need to build indices or other data structures. It may also require substantial spare storage capacity for temporary files. Therefore a *terminology server* provider may choose to pre-import the distribution files and provide them to users in pre-prepared form. However, an import facility should also be available in a suitably secured form to end-user organizations, to enable installation and maintenance of *Extensions*.

4.2.2 Importing new releases and updates

This section is concerned with updates to the *SNOMED CT* resource accessed by the *terminology server*. A separate section addresses issues related to management of *SNOMED CT* encoded records, queries and protocols following a new *SNOMED CT release*.

4.2.2.1 Update Release Formats

In *Release Format 1*, each release of *SNOMED CT* content is distributed as a complete snapshot - rather than as a set of changes and additions. Additional history information is provided in files described in the *Content History*.

Note: *Release Format 2* has a different approach to *release files* and history (see *Importing SNOMED CT release data* and *History Mechanism*).

4.2.2.2 Updating from distribution files

Updating a *terminology server* from a new release of distribution files is similar to the original import. However, the import process should support the retention and/or reinstatement of local configurations such as installed *Extensions* and selected *Subsets*.

An alternative to repeating the full import process is to read and apply the changes indicated in the *Component History Table*. The distribution files are still required as the source for additional components that have been added between releases. Also, the *Component History Table* only indicates changes to *Concepts* and *Descriptions*. However, changes to the distribution format to allow better representation of all *SNOMED CT component* changes is under discussion.

This process is outlined as follows:

- Read the *Component History Table*:
  - Identify all changes since the last release used to update this server.
- For each of these changes check the *ChangeType* value:
  - If the *ChangeType* is 1 (*status* change), apply the new *status* to the component in the *terminology server* resource:
    - For example, if the new *status* of a *Concept* is Duplicate, this *status* is applied to the identified *Concept*;
    - For many inactivated *concepts*, the *Relationships* file can be used to identify an *active concept* that may be a candidate to use in place of the inactivated *concept*.
  - If the *ChangeType* is 0 (added), find the relevant component in the distribution files and add this to the *terminology server* resource.
  - If the *ChangeType* is 2 (minor change), find the relevant component in the distribution files and apply the new data to the component in the *terminology server* resource.

This approach is more complex than importing the new release. However, it may make it easier to retain local configuration data (e.g. selected *Subsets* and any *Extensions* previously installed).

If a *terminology server* suppliers pre-imports *SNOMED CT releases* for distribution to their users, this approach facilitates the production of re-distributable updates.

4.2.2.3 Changes to Relationships

Unlike other *core files* the released *Relationships Table* has no *status* field and only *active Relationships* are distributed. Therefore *Relationship* changes are not recorded in the *Component History Table* and requires a full import of this table or the need for an intermediate table to determine which *Relationships* were added and removed in the distribution.

4.2.3 Importing Extensions

Terminology servers should be able to import *Extensions*.

The process of importing an *Extension* is similar to importing the main distribution files. However, some additional functionality is required to ensure appropriate installation, maintenance and use of *Extensions*. Applications should:

- Allow the users or user communities to specify the *Extensions* to be recognized by their systems. Before recognizing any *Extension*, users should check that:
  - The *Extension* has been supplied by the *IHTSDO* or another organization authorized by the *IHTSDO* to provide such *Extensions*.
  - You are satisfied with the quality control procedures of the providing organization :
    - Authorization of an organization to produce *Extensions* does not imply any seal of approval related to the quality of *Extensions* provided by those organizations ;
    - Installation of *Extensions* is done entirely at the risk of the user subject to their license agreement with the provider of the *Extension* and/or the application developer.
- Check each *Extension* prior to installation to ensure that:
  - It is one of the *Extensions* recognized by the user.
  - Any dependencies of the *Extension* have been met. These dependencies may include:
    - Installation of a particular *SNOMED CT release*;
    - Prior installation of other *Extensions*.
  - The installation procedure has pre-checked all *components* in the *Extension* to ensure that:
    - All ComponentIDs:
      - Are unique;
      - Have *partition identifier* and *namespace-Identifier* values appropriate to the provider of that *Extension*
      - Have a valid *check-digit*
    - All fields meet data type, size and value *constraints* specified for the relevant tables.

Caution:

If any *components* fail any of these tests the entire *Extension* must be rejected.

- Import *Extension* files in a way that replaces earlier versions of the same *Extension* but add to, rather than replacing, the *SNOMED CT core* distribution and other installed *Extensions*;
- Reject, highlight or apply other agreed business rules to information received by the system that contains *SCTIDs* for *components* from *namespaces* that are not in the list, or recognized *Extensions*.

4.2.4 Providing access to history information

When a new release is installed, an application providing *record services* may need to update records, indices, queries, or protocols to take account of changes to the terminology.

Caution: Updates to records as a result of terminology version changes should be limited to addition of updated information. The integrity of original clinical record entries must not be compromised by any updating process. The original entry should be retained and should be reused to support subsequent updates.

To support these maintenance activities a *terminology server* should provide access to:

- Information in the *Component History Table* indicating which components have been changed since the previous release.
- Information in the *References Table* identifying components which replace or were duplicated by *inactive components*:
  - Not applicable to *Concepts* (see next bullet below).
- Information in the *Relationships Table* identifying *Concepts* that:
  - Replace an "erroneous" *Concept*:
    - RelationshipType | REPLACED BY |
  - Are identical to a "duplicate" *Concept*:
    - RelationshipType | SAME AS |
  - Are potential meanings of an "ambiguous" *Concept*:
    - RelationshipType | MAY BE A |
  - Were previously *subtypes* of an *Inactive Concept*:
    - RelationshipType"WAS A" (Future Use).

Access to the *Component History Table* and *References Table* may only be required to support for a time a limited set of maintenance activities following each release. However, access to information about *Active Concepts* related to *Inactive Concepts* is often required for day-to-day operation of the application. For example, it will affect the selective retrieval and analysis of data information originally recorded with a *Concept* that is now *inactive*.

4.3 Foundation Terminology services

The following list summarizes a set of essential services that any *terminology server* is likely to require. Some of these are reiterated in more detail in subsequent sections. In other cases, support for a more complex service presumes the existence of one or more of these foundation services.

4.3.1 Access to release information

Terminology servers should enable client applications and users to access the *current SNOMED CT release version information*.

4.3.2 Access to components

4.3.2.1 Access to concepts

A *terminology server* should enable client applications to rapidly find any *Concept* by any of the following criteria:

- *ConceptId*;
- *SNOMEDID*
- *CTV3ID*

Once a *Concept* has been found, the client application should be able to read the values. Any of the properties of that *Concept* included in the *Concepts Table* should be accessible.

4.3.2.2 Access to Descriptions

A *terminology server* should enable client applications to rapidly find any *Description* or set of *Descriptions* by any of the following criteria:

- *DescriptionId*;
- *ConceptId*;
- ConceptIdand *DescriptionType*

Once a *Description* has been found the client application should be able to read the values of any of the properties of that *Description* included in the *Descriptions Table*.

4.3.2.3 Access to Relationships

A *terminology server* should enable a client application to rapidly find any *Relationship* or set of *Relationships* by any of the following criteria:

- *ConceptId1*;
- *ConceptId1*, *CharacteristicType* and RelationshipType;
- *ConceptId1*, *CharacteristicType*, *Relationship group* and RelationshipType;
- *ConceptId2*;
- *ConceptId2*, *CharacteristicType*, and RelationshipType.

Once a *Relationship* has been found the client application should be able to read the values of any of the properties of that *Relationship* included in the *Relationships Table*.

4.3.3 Access to essential concept Identifiers

Terminology servers should provide efficient access to the *Identifiers* that represent *concepts* with structurally significant *Roles* within the terminology. *Table 235* lists the *concepts* that have the most clear-cut structurally significant *Roles*. A *terminology server* should enable access to these *Identifiers* by an easy to use name of enumeration. In addition a *terminology server* should provide a service that rapidly determines whether a given *concept* is a *subtype* of any of these *concepts*. It is also useful to for the *terminology server* to extend similar functionality to all direct *subtypes* of the *root concept* (| SNOMED CT Concept |) and to *subtype descendants* of | concept model attribute |.

Table 235: Essential concept Identifiers

Id	*Preferred Term*	Significance
138875005	| SNOMED CT Concept |	The *root concept*. All other *active concepts* are *subtypes* of this *concept*.
900000000000441003	| SNOMED CT Model Component |	All active metadata *concepts* are *subtypes* of this *concept*.
900000000000442005	| core metadata concept |	All enumerated values applicable to core *components* are *subtypes* of this *concept*.
900000000000454005	| foundation metadata concept |	All *reference sets* and all *reference set*-related metadata *concept* are *subtypes* of this *concept*.
900000000000455006	| reference set |	All *reference sets* are *subtypes* of this *concept*.
116680003	| is a |	The *Attribute* used to specify the *subtype Relationship* between *concepts*.
246061005	| attribute |	All *Attribute* (*relationship type*) *concepts* are *subtypes* of this *concept*.
410662002	| concept model attribute |	With the exception of the *subtype Relationship* (see above) all *relationship types* that are used in the *SNOMED CT Concept Model* are *subtypes* of this *concept*.
370136006	| namespace concept |	Each *subtype* of this *concept* represents an extension namespaces allocated by the *IHTSDO*.
363743006	| navigational concept |	*Subtypes* of this *concept* to provide nodes in *navigation hierarchies*. They act as grouper categories that do not have any semantic meaning and thus do not appear elsewhere in the *SNOMED CT* hierarchy.

Id	***Preferred Term***	**Significance**
410663007	| concept history attribute |	In *Release Format 1* the *subtypes* of this *concept* specify types of *historical relationships*. Not used in *RF2*.
362955004	| inactive concept |	In *Release Format 1* all *inactive concepts* are *subtypes* of one of the *subtypes* of this *concept*. Not used in *RF2*.
370115009	| special concept |	In *Release Format 1 concepts* that are not intended for use as content are *subtypes* of this *concept*. Not used in *RF2* as it is replaced and extended by | SNOMED CT Model Component | (see above).

4.4 User Interface Terminology services

This section of the guide is concerned with *Terminology services* that allow users to view and select of *SNOMED CT Concepts* and *Descriptions*.

4.4.1 Text Searches

Effective implementation of *SNOMED CT* depends on the speed and simplicity with which users can locate the *terms* and *concepts* that they wish to use. A busy clinical user may become frustrated if the content they need cannot be quickly located when they search using familiar words or phrases. For this *reason* an efficient search strategy should address the following issues:

- Speed of search:
  - Search speed should be optimized by use of appropriate indexes.
- Search should not be too sensitive to word *order* or exact phrasing:
  - Search should be insensitive to word - *order* variants:
    - For example, "head pain" for | pain in head |
  - Allow use of acronyms or abbreviations for frequently used *terms*:
    - For example, "MI" for "myocardial infarction" or "mitral incompetence".
  - Search should take account of word form variants:
    - For example, "inflamed", "inflammatory", "inflammation".
- Excessive search results should not hinder selection of the required *concept*:
  - When several *synonyms* of the same *concept* match the search key, only one should be displayed.

The purpose of this section of the implementation guide is to describe strategies a developer might use to implement the search requirements outlined above.

The *SNOMED CT* Developer Toolkit contains several files, which help to support efficient search mechanisms. These include the *Excluded Words Table*, four *keyword* indexes and the *Word Equivalents Table* summarized by *Table 29* and *Table 30*.

4.4.1.1 Single keyword index

The single *keyword* table, (DescWordKey), provides a pointer from each *keyword* used in any *Description*, to the *Descriptions* in which that *keyword* is used. The purpose of the single *keyword* index is to support a search capability, which is independent of the *order* in which words appear in a *description*. The single *keyword* index represents the minimum necessary supporting structure for searches on *SNOMED CT* content. Searches involving target words that appear in many *descriptions* may be unacceptably slow if searches are carried out using the single *keyword* index alone. Developers wishing to produce applications with faster search times are encouraged to supplement their system with a multiple *keyword* index such the DescDualkey table (see *Word Search Tables*) provided as part of the *SNOMED CT release*.

Note that some words that are used in *description* are linking words, which are unlikely to be in the target of a search. These words are not considered to be *keywords* and may be excluded from the *keyword* index. They are found in *Excluded Words* File.

4.4.1.1.1 Generating the single keyword index

Although single *keyword* indexes are available in the Developer Toolkit for the *International Release* English *language descriptions*, developers may need to generate *keyword* entries for any *descriptions* added as part of an *Extension*. A toolkit is available and this contains the programs used to generate the *International Release keyword* indexes.

Entries may be added to the single *keyword* table by following the method outlined below.

For each *description*, parse the text of the *term*:

- To avoid inappropriate case mismatches, convert all characters to the same case.
- Extract words by breaking at spaces, punctuation marks, and brackets.
- For each word:
  - If the word is not in a list of *excluded words*, add a row to *keyword* table.

4.4.1.1.1.1 Example: Generation of keywords for a sample Description

Table 236: Sample Description

DescriptionId	*ConceptId*	*Term*
22565018	13185000	| pyrogallol 1,2-oxygenase |

- Convert all characters to the same case.

"Pyrogallol 1,2-oxygenase" -> "PYROGALLOL 1,2-OXYGENASE"

- Extract words by breaking at spaces, punctuation marks, and brackets.

"PYROGALLOL 1,2-OXYGENASE" -> ' (1) = "PYROGALLOL"

(2) = "1"

(3) = "2"

(4) = "OXYGENASE"

- For each word:

- If the word is not in a list of *excluded words*, and length of word > 1, and first character is not numeric:
    - Add a row to *keyword* tables;
    - Only the first eight characters are used in the *keyword*.

Table 237: DescKey Words

KeyWord	*DescriptionId*
PYROGALL	22565018
OXYGENAS	22565018

Table 238: ConcKeyWords

KeyWord	*ConceptId*
PYROGALL	13185000
OXYGENAS	13185000

4.4.1.1.2 Search using the single keyword index

A single *keyword* search may be conducted as follows:

- The user-typed search *string* is converted to consistent case;
- The *string* is parsed, breaking at spaces and punctuation characters;
- One word is selected from the parsed word list to use as a look-up on the single *keyword* index;
- Look-up on the single *keyword* index may be "exact" or "starts with," depending on wild card conventions used in the search *string*.

4.4.1.1.2.1 Example: Search using single key-word index

The user searches for "Hip* replacement*" (where "*" represents the wild card for any number of extra characters).

- The user-typed search *string* is converted to consistent case.

"Hip* replacement" -> "HIP* REPLACEMENT*"

- The *string* is parsed, breaking at spaces and punctuation characters.

"HIP* REPLACEMENT*" -> (1) "HIP*"

(2) "REPLACEMENT*"

- Look up "HIP" on the single *keyword* index using "starts with" *query*.

Table 239: Example results for a Search for "hip"

Count	*DescriptionId*	*ConceptId*	*Term*
1	49926016	29836001	| hip |

Count	*DescriptionId*	*ConceptId*	*Term*
2	196344018	24136001	| hip |
3	2296013	736004	| abscess of hip |
4	1480791012	386649003	| partial hip replacement by prosthesis |
. . . . .	. . . .	. . . .	. . . .
315	371616001	1210239015	| methenamine hippurate 1g tablet |

Using a *DualKey* index is more efficient as the same search finds only 11 matches.

Table 240: Sample results of a search for "hip replacement" using DualKey "HIPREP"

Count	*DescriptionId*	*ConceptId*	*Term*
1	1480791012	386649003	| partial hip replacement by prosthesis |
2	33592011	19954002	| total replacement of hip with use of methyl methacrylate |
3	50150016	29969002	| replacement of acetabulum of hip |
4	54398014	32581000	| partial hip replacement by cup with acetabuloplasty |
. . . . .	. . . .	. . . .	. . . .
11	183737015	112728000	| total revision of hip replacement with use of methyl methacrylate |

4.4.1.2 Multiple keywords

The performance of single *keyword* searches is highly dependent on the number of candidate *descriptions* returned by the *keyword* for subsequent filtering. The extremely high number of matches for some words in common use makes it likely that some searches will be unacceptably slow.

One way to alleviate this problem would be to create a table containing a row for all combinations of word pairs in each *description*. In some database environments that support optimization of multiple key searches, this may offer no benefits. However, in other environments, such a table may substantially speed searches.

A comprehensive word pair table would be very large. Such a table covering the full content of *SNOMED CT* would contain approximately 1.5 million unique word pairs and 6 million rows. Limiting the unique keys to the first three letter of each word reduces the table size to a more readily optimized set of keys. This requires the final part of the search to be conducted using text comparison (since the keys are incomplete).

4.4.1.2.1 Generating the DualKey index

Although *Dualkey* indexes are available as part of the Developer Toolkit , it is important to know how this table is generated. *SNOMED CT* users that generate *Extensions* should follow the method outlined below to generate new entries in the *Dualkey* index, based on the *descriptions* in the *Extension*.

For each *description*, parse the text of the *term*:

- To avoid inappropriate case mismatches, convert all characters to the same case;
- Extract words by breaking at spaces, punctuation marks, and brackets;
- For each word of three characters or more that is not in the list of *excluded words*, extract the first 3 characters, and arrange the word fragments in alphabetical *order*;
- Generate the dual keys for this *description* by concatenating each word fragment with those that come after it in the list;
- For each dual key, add a row to the word pair tables.

4.4.1.2.1.1 Example: Generation of keywords for a sample Description

Table 241: Sample Description

DescriptionId	*ConceptId*	*Term*
33592011	19954002	| total replacement of hip with use of methyl methacrylate |

- To avoid inappropriate case mismatches, convert all characters to the same case.

"TOTAL REPLACEMENT OF HIP WITH USE OF METHYLE METHACRYLATE"

- Extract words by breaking at spaces, punctuation marks, and brackets.

1. TOTAL;
2. REPLACEMENT;
3. OF;
4. HIP;
5. WITH;
6. USE;
7. OF;
8. METHYLE;
9. METHACRYLATE.

- For each word of three characters or more, that is not in the list of *excluded words*, extract the first 3 characters, and arrange the word fragments in alphabetical *order*.

1. HIP;
2. MET;
3. REP;
4. TOT;
5. USE.

Note:

"OF" is less than 3 characters and is an *excluded word*, "WITH" is an *excluded word* and "MET" is duplicated, so we only include it once.

- Generate the dual keys for this *description* by concatenating each word fragment with those that come after it in the list;
- For each dual key, add rows to the word pair tables.

Table 242: DescDualKey

Dual key	***DescriptionId***
HIPMET	33592011
HIPREP	33592011
HIPTOT	33592011
HIPUSE	33592011
METREP	33592011
METTOT	33592011
METUSE	33592011
REPTOT	33592011
REPUSE	33592011
TOTUSE	33592011

Table 243: ConcDualKey

Dual key	***ConceptId***
HIPMET	19954002
HIPREP	19954002
HIPTOT	19954002
HIPUSE	19954002
METREP	19954002
METTOT	19954002
METUSE	19954002
REPTOT	19954002
REPUSE	19954002
TOTUSE	19954002

4.4.1.2.2 Searching for Descriptions using the DualKey index

A search on the dual key index can only be carried out if the user enters a search *string* that contains at least two word fragments both of which are three characters or more in length. If the search *string* does not meet this criterion, the single *keyword* search mechanism must be used.

- The user-typed search *string* is converted to consistent case;
- The *string* is parsed, breaking at spaces and punctuation characters;
- For each word of three characters or more, extract the first 3 characters, and arrange the word fragments in alphabetical *order*;
- Create a dual key by concatenating the first two 3 letter word fragments;
- Use this dual key to look up exact matches on the word pair index;
- *Descriptions* found by searching on the word pair index are screened, to see if they contain the complete words in the original search *string*

4.4.1.2.2.1 Example: Search using word pair index

User searches for "PYRO* 1 OXYGEN*".

- The *string* is parsed, breaking at spaces and punctuation characters.

1. "PYRO*";
2. 1;
3. "OXYGEN*".

- For each word of three characters or more, extract the first 3 characters, and arrange the word fragments in alphabetical *order*.

1. "OXY";
2. "PYR".

- Create a dual key by concatenating the first two 3 letter word fragments.

OXYPYR

- Use this dual key to look up exact matches on the word pair index.

Table 244: Sample results of a search for "PYRO* 1 OXYGEN*"

Dual key	***DescriptionId***	***Description***
OXYPYR	1969019	| o-Pyrocatechuate oxygenase |
OXYPYR	22565018	| pyrogallol 1,2-oxygenase |
OXYPRY	104951019	| 2,5-Dihydroxy-pyridine oxygenase |

- *Descriptions* found by searching on the word pair index are screened, to see if they contain the complete words in the original search *string* :
  - *Description* 1969019 is eliminated since it does not contain the word "1";
  - Description104951019 is eliminated, it does not contain the word "1" or any word beginning with the *string* "pyro".

4.4.1.3 Using word equivalents to enhance searches

In healthcare, there are many words with equivalent meanings. *Synonyms* provide alternative phrases referring to the *concept*. However, *synonyms* are not created automatically for every possible combination of words with an equivalent meaning. The success of simple searches using one or more *keywords* depends on the text of the available *descriptions*. Therefore searches will fail or will be incomplete where a different equivalent word is used in the search.

For example: "Kidney stone" and "Renal calculus" are synonymous *descriptions* in *SNOMED CT*. A search of *SNOMED CT* for the target phrase "kidney stone fragmentation" yields the result "Percutaneous nephrostomy with fragmentation of kidney stone," while a search for "Renal stone fragmentation" yields no results.

One way of addressing this problem is to maintain a table of *word equivalents*. A table of this type is a prerequisite for exhaustive *synonym* generation. An initial set of *word equivalents* is included in the *SNOMED CT* Developer Toolkit. Individual implementers will wish to add additional *word equivalents* to meet the requirements of their particular medical specialty or user needs. This table is an additional resource to assist searching and parsing of phrases. It need not be a comprehensive dictionary of words. Many searches can be completed without reference to this table so it need not contain every word or equivalent phrase used in *SNOMED CT*.

Several factors complicate the initial population and subsequent use of the *word equivalents table*:

- A phrase of two or more words may be equivalent to a single word.

Example:

"Endoscopic esophagus examination" is equivalent to "esophagoscopy"

- A word may have more than one meaning, and in this, only one meaning of a pair of words may be equivalent. Thus an apparent enhancement of a search may in practice lose some of the specificity of the intended search.

Example:

"Tap" and "aspiration" are equivalent in the context of *terms* such as "pleural tap", "pleural aspiration", but not in the context of a "patella tap", a physical "tap" on a bag or catheter, or the clinical disorder "neonatal aspiration syndrome."

- When searching using incomplete words and/or wildcards, use of *word equivalents* may impede effective searches by increasing the number of spurious potential matches. This either extends the processing required to filter the real matches from the potential matches or increases the length of the list of choices presented to the user.

A wise system developer will allow the user to customize their search options, enabling searches to be narrowed, or extended to meet the needs of varying circumstances.

4.4.1.3.1 Example: Using word equivalents table to extend a failed search

A system user enters the search *string* "Fragmentation of renal calculus;" the search returns no results. The search application that the user has been provided with has the option to extend the search by using the *word equivalents table*. The user selects this option and searches again using the same search *string*.

The *word equivalents table* contains the following relevant entries:

Table 245: Example Rows from Word Equivalents Table

WordBlockNumber	*WordText*	*WordType*
1021	KIDNEY	2 (*word equivalent*)

WordBlockNumber	***WordText***	***WordType***
1021	RENAL	2 (*word equivalent*)
4430	CALCULUS	2 (*word equivalent*)
4430	CALCULI	1 (word form variant)
4430	STONE	2 (*word equivalent*)
9870	RENAL STONE	4 (equivalent phrase)
9870	KIDNEY STONE	4 (equivalent phrase)
9870	KIDNEY CALCULUS	4 (equivalent phrase)
9870	RENAL CALCULUS	4 (equivalent phrase)
9870	NEPHROLITH	2 (*word equivalent*)

The table is used to make substitutions in the search *string* to produce all possible unique search variants:

"Fragmentation of renal calculus"

"Fragmentation of renal stone"

"Fragmentation of kidney stone"

"Fragmentation of kidney calculus"

"Fragmentation of Nephrolith"

"Fragmentation of renal calculus"

"Fragmentation of renal calculi"

"Fragmentation of kidney calculi"

These 8 search *strings* are used as the target phrase for *keyword* searches on the word pair index. Results from all 8 searches are combined, and duplicate *concepts* are eliminated, giving the final list of search results.

4.4.1.4 Rationalizing searches that return duplicate hits

In the previous sections of this guide, we have considered methods of ensuring that searches on a target phrase maximize the possibility of finding the *concept* that the system user requires. It is equally important to prevent the search results from containing excessive matches, since these will require filtering by the user, imposing an additional burden. Some strategies for limiting the number of search results displayed are discussed below.

4.4.1.4.1 Avoiding multiple hits on the same concept

In many instances several *synonyms* associated with the same *concept* contain the same *keyword*. The designer of search software may consider filtering the output of search results so that only the first matching *description* for a *concept* is displayed.

Example:

"Endoscopic examination of the stomach" and "endoscopy of the stomach" are *synonyms* of the same *concept*. A search for the target phrase "endo* stomach" would return the first phrase found during the search. The second would be excluded, since it has the same *concept identifier* as an existing match for this search.

4.4.1.4.2 Constraining and extending search parameters

User configurable options may be one way of limiting search results. Three possible methods of limiting search results through user configurable options are suggested here:

- Limiting searches to exact matches unless wild cards are used. A search on a single word may produce many matches if it is assumed that the user is searching for any phrase that contains the target word. Forcing the use of wild cards for this kind of search can help avoid this problem.
- Make searches that include use of " *word equivalents*" a user configurable option that can be used to extend or constrain a search.
- Display search results a few at a time, with most frequently used *descriptions* listed first. This option will require the application to track the frequency of *term* selection so that search results can be sorted in this way.

4.4.2 Hierarchical Navigation

This section of the guide describes the *Terminology services* that are likely to be required to navigate *SNOMED CT* hierarchies.

One of the key strengths of *SNOMED CT* is a rich set of *relationships* that connect the *concepts* within the terminology. The primary use of these *relationships* is to facilitate selective retrieval. However, some of these *relationships* are arranged in hierarchies that can be navigated using an appropriate user-interface control. For example, the *subtype hierarchy* formed by the |is a| *relationships* can be used to navigate from a selected *concept* to a more or less specific *concept*.

SNOMED CT also specifies ways in which alternative *navigation* hierarchies can be created. Unlike *relationship* based hierarchies, *navigation* hierarchies convey no semantic information but are intended to be used to enhance the user experience when navigating through the terminology.

4.4.2.1 Access to hierarchically related concepts

Terminology servers should enable client applications to access collections of *Concepts* that are related to a specified *Concept* as:

- *Subtype children*
- *Subtype descendants* (includes all generations of *children*);
- *Supertype parents*
- *Supertype ancestors* (includes all previous generations of parents).

4.4.2.2 Using |is a |Relationships for hierarchy navigation

4.4.2.2.1 What is the SNOMED CT hierarchy?

The "*SNOMED CT hierarchy*" refers to the organization of *concepts* in *SNOMED CT* from the general, at the top of the *hierarchy*, to the more specific or "granular" at the bottom. The *concepts* that make up the very top level of the *hierarchy* are shown in *Table 246*. All other *SNOMED CT concepts* fall under one or more of these categories.

Table 246: Top Level *Concepts*

• *| Clinical finding |*	• *| Physical force |*
• *| Procedure |*	• *| Event |*
• *| Observable entity |*	• *| Environment or geographical location |*
• *| Body structure |*	• *| Social context |*
• *| Organism |*	• *| Situation with explicit context |*
• *| Substance |*	• *| Staging and scales |*
• *| Pharmaceutical / biologic product |*	• *| Physical object |*
• *| Specimen |*	• *| Qualifier value |*
• *| Special concept |*	• *| Record artifact |*
• *| Linkage concept |*	

Several levels of increasingly fine categorization may exist between the top level of the *hierarchy* and *concepts* that have sufficient detail to be recorded in a patient's medical record. *Figure 45* shows the levels of *hierarchy* that exist between the top-level *Concept* "Finding/disorder" and the finding "Catatonic reaction."

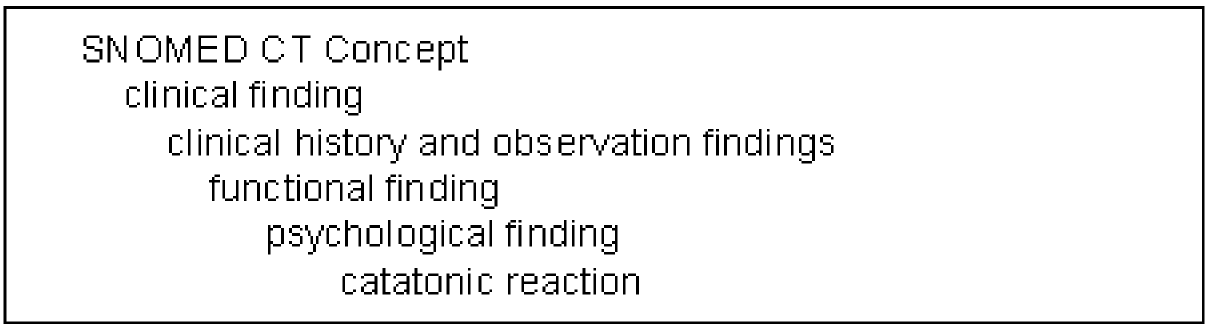


Figure 45: Hierarchy example: Catatonic Reaction

4.4.2.2.2 How is hierarchical information represented in the relationships table?

The *SNOMED CT relationships table* represents *Relationships* between one *SNOMED CT concept* and another by including a row in the table for each such *Relationship*. The columns *ConceptId1*, *Relationship Type* and *ConceptId2* define the source of the *Relationship*, the kind of *Relationship* that exists and the target of the *Relationship* respectively. Each of these fields, including *Relationship Type*, is a *SNOMED CT concept Identifier*. Hierarchical *Relationships* are expressed by linking the source *concept* to its "parents" i.e. the *concept* or *concepts* immediately above it in the hierarchy. The *Relationship Type* used to make this hierarchical link is known as the | is a | *Relationship*.

For example, we can say | Catatonic reaction | | is a | | Psychological finding |, this would be expressed in the *relationships table* as follows:

Table 247: Relationship Example

ConceptId1	*Relationship Type*	*ConceptId2*
102909009	116680003	116367006

Where:

- 102909009 is the *concept identifier* for | Catatonic reaction |;
  ;
- 116680003 is the *concept identifier* for the | Is a | *Relationship*;

;

- 116367006 is the *concept identifier* for | Psychological finding |.

.

Conversely, by inverting the | is a | *Relationship* we can find the children of the target *Concept*, i.e. the *Concept* or *Concepts* immediately below it in the hierarchy.

4.4.2.2.3 Using | is a |Relationships to enhance search capabilities

This section is concerned with the ways in which the *hierarchy* can be used to help a *SNOMED CT* user when they are searching or browsing the terminology.

Note: The primary use of the *SNOMED CT subtype hierarchy* is to support effective retrieval and aggregation of data. This is discussed elsewhere in *Testing and traversing subtype relationships*.

It is possible to start at the top of *hierarchy* and navigate from parent to *child* in order to find a *Concept* or *term* in *SNOMED CT*. A more efficient approach, however, is to use the *hierarchy* to supplement a *keyword* search by enabling the user to look at related *Concepts* in order to consider them as alternative matches, or to check the context of a search result. The examples below illustrate these two uses of the *SNOMED CT hierarchy*.

4.4.2.2.4 Example: Using the hierarchy to check context

A user wishes to find a *description* that relates to the condition of a patient who is hypersensitive to an allergen. The user performs a search on the *keyword* "Hypersensitivity" and finds an exact match. Before the user selects the *description* for inclusion in the patient record, they check the *fully specified name*, which is "Sensitivity (finding)." The user then checks the *hierarchy* and discovers that the selected *Concept* has "Psychological finding" as an *ancestor*, which indicates that this is not the correct *description* to use in this context.

4.4.2.2.5 Example: Using the hierarchy to check related Concepts

A user wishes to find a *description* that relates to the condition of a patient who is hypersensitive to an allergen, as in the example above. This time, a different approach is taken. The user searches for the *keyword* "allergy," and finds one *Concept* having a *description* that is an exact match. The user then looks at the *children* of the *Concept*, i.e. those *concepts* immediately below it in the *hierarchy*. One of the *children* has the preferred *description*"Contact Hypersensitivity" which matches the user's intended meaning. The user selects this *Concept* for inclusion in the patient record.

4.4.2.2.6 Using | is a |Relationships to display hierarchical information in applications

Most visual application development tools contain a *component* designed to display hierarchical information as a tree in which branches can be expanded or collapsed. Tree views are well-suited to displaying *SNOMED CT* hierarchical *Relationships* (see *Figure 46*). These views are used in many different user-interfaces where information needs to be represented as a hierarchy (e.g. displaying a file-system as a hierarchy of folders or providing a collapsable outline of a document or help file). Therefore, most users will already be familiar this paradigm.

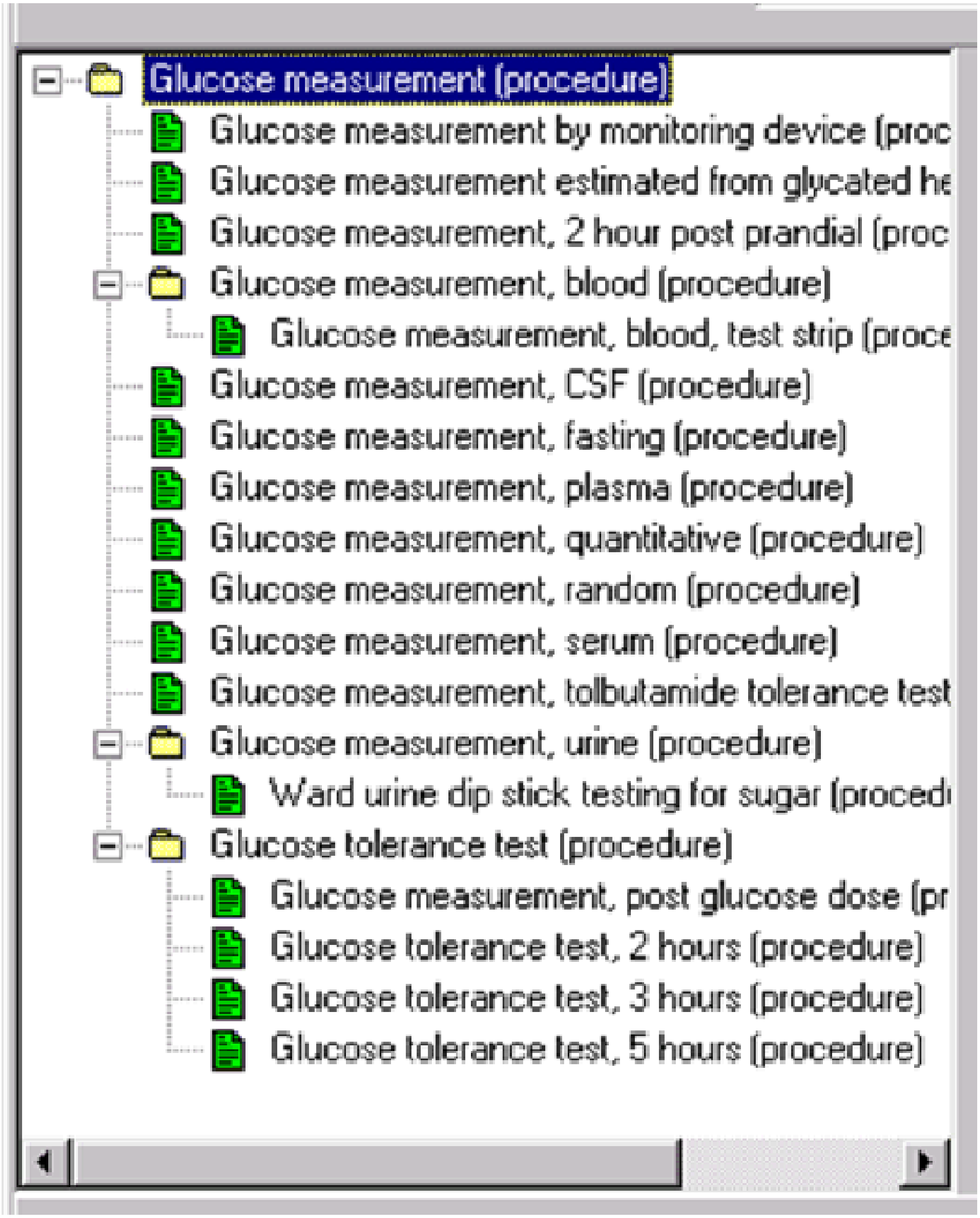


Figure 46: SNOMED CT hierarchy represented in a tree view

The process of creating a tree view from the *SNOMED CT Relationship* table is straightforward as long as a few simple ideas are mastered:

- Most standard tree-views controls start from a single root and require that higher level branches must be added before sub-branches. This means that when viewing part of the *hierarchy* from the bottom up, the tree must be compiled in temporary form before it can be displayed.

- Since the depth of the *hierarchy* is not known in any particular case, operations that iterate up or down the depth of the *hierarchy* must be done using a recursive algorithm. However, this recursion must usually be limited since placing the entirety of the *SNOMED CT* hierarchy in a single tree control is likely to create performance issues and may exceed physical limits on the capacity of the control.
- Standard tree view controls are not good at displaying the multiple parent nodes that occur in a *polyhierarchy* like *SNOMED CT*. Therefore, some compromises need to be made to present options for navigation up the hierarchy.
- Effective use of some tree controls requires unique keys for each node. Multiple parents and multiple roots through the hierarchy mean that the same *Concepts* will appear in multiple places in the hierarchy. Therefore, the *concept identifier* cannot be used to provide a key that is globally unique within the hierarchy.

4.4.2.3 Using "Part of" Relationships for hierarchy navigation

In addition to the *subtype hierarchy* represented by | is a | *relationships*, *SNOMED CT* also represents a partonomy hierarchy using | Part of | *relationship*. This creates an alternative hierarchy which can be also be used for navigation. The difference between these hierarchies is that:

- The *subtype hierarchy* relates body structure *concepts* to supertypes that represent *the whole or any part of* an organ or other body part. The *fully specified names* of these supertype *concepts* include the word 'structure'.

  Example: | Right ventricular structure | (is) | is a | | heart structure |

- The partonomy hierarchy relates body structure to *concepts* to *concept* that represent | the entirety of | or an organ or anatomical structure of which they form part

  Example: | Entire right ventricle | (is) | part of | | entire heart |

4.4.2.4 Using other Relationships to navigate SNOMED CT content

Many *SNOMED CT Concepts* have *relationships* with content in other areas of terminology. These *Relationships* are one of the ways in which *SNOMED CT* provides computer readable definitions for medical *concepts*. For example, diseases in *SNOMED CT* generally have a *Relationship* to the body site affected by the disorder and a *Relationship* to the morphology associated with the disease. Procedures in *SNOMED CT* might have *Relationships* to the *concept*, which defines the type of surgical action being carried and the procedure site, for example. Examples of *Relationships* for a disease and a procedure are shown below. A full list of the *Relationships* that can be used for each type of *Concept* can be found in *Table 3*.

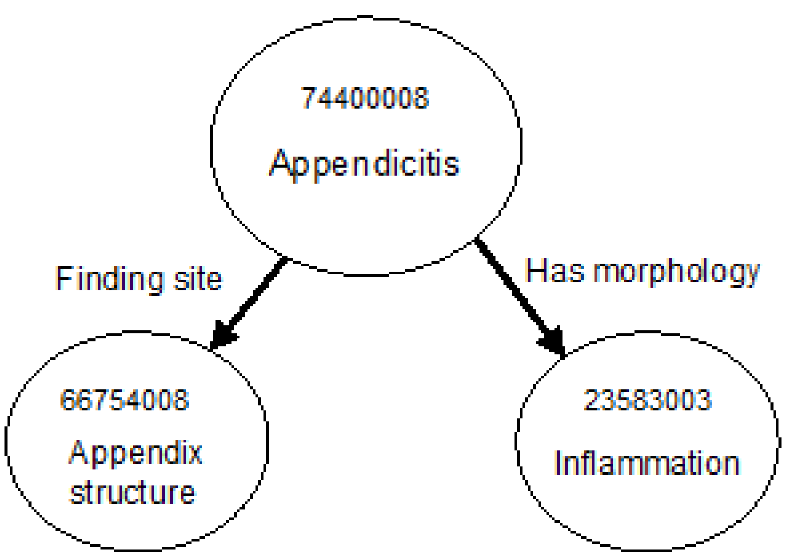


Figure 47: Relationship for disease appendicitis

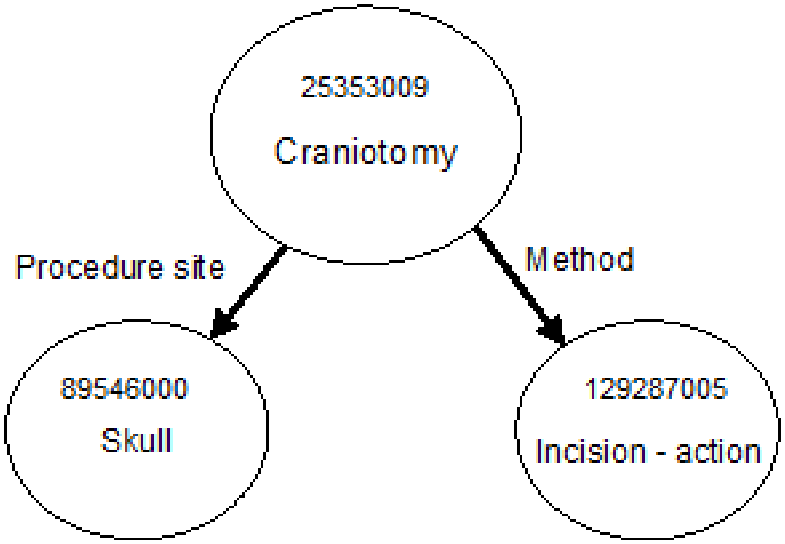


Figure 48: Relationships for procedure craniotomy

These *Relationships* are very useful in the context of data retrieval and analysis. The *Relationships* can also be used to aid in the search for specific *SNOMED CT Concepts* in cases where the *term* alone may not sufficiently distinguish between choices. For example, a search for all inflammatory diseases of the lung could be carried out as follows:

- Use the *hierarchy* to compile a list of all *Concepts* that are lung structures;
- Search for any *Concept* that has a row in the *Relationships table* with a *Relationship Type* of "Disorder site," and with *ConceptId2* included in the list of lung structures;
- Now exclude any procedures from the list that do have the | Associated morphology | "Inflammation" in the *Relationship* table;
- Final product is a list of all lung disorders that involve inflammation.

To achieve these same results with a *string* search we would have to perform separate searches for | pneumonia |, bronchitis, | pleurisy | and many other conditions that cannot be linked via a sample *string* search.

4.4.2.5 Using navigation Subsets to display application-specific hierarchies

The file structure of *Subsets* is described in detail in the *Subset Mechanism* specification. This section explains how *navigation Subsets* may be used to create and display a customized *hierarchy* for use in an application.

4.4.2.5.1 Creating a navigation subset

Let us take the example of a hospital specializing in the diagnosis and treatment of cancer. For the purposes of the application, the designers wish to have two top-level categories: "Diagnosis," including only diagnoses relating to the specialist area; and "Interventions," including subcategories for chemotherapy, radiation, and surgical treatments.

First we would find the *Concepts* that we wish to go into the *navigation hierarchy*. A *Subset* can only include existing *Concepts*, by definition, so if any of the *Concepts* that we require do not exist in the *SNOMED CT core files*, they will have to be added, either by requesting new *concepts* from the *IHTSDO*, or via an approved *Extension*. For the sake of our example, let us assume that all the required *Concepts* are part of the *SNOMED CT core concepts*. The set of *Concepts* is shown in *Table 248*.

Table 248: Sample Concepts for Navigation Subset

concept identifier	***Preferred Term***
309902002	| clinical oncology department |
55342001	| neoplastic disease |
362956003	| procedure / Intervention |
108290001	| radiation oncology AND/OR radiotherapy |
69960004	| antineoplastic chemotherapy regimen |
387713003	| surgical procedure |

In order to complete the *navigation Subset* we must add entries to the relational *Subsets table*, and to the *Subset Members Table*. For each new *Subset* an entry is required in the *Subsets Table*. The *Subset Members Table* uses the field LinkedID to specify the navigational *children* for each entry. The member *status* specifies the display order for the *children*.

Table 249: Subsets Table Example

Field Name	**Value**
SubsetId	309902039
SubsetOriginalID	
SubsetVersion	1.0
SubsetName	Clinical Oncology Department
SubsetType	*Navigation*
LanguageCode	
RealmID	
ContextID	

Table 250: Subset Members Table Example

SubsetId	**MemberID**	***MemberStatus***	**LinkedID**
309902039	309902002	0	55342001
309902039	309902002	1	362956003

SubsetId	MemberID	*MemberStatus*	LinkedID
309902039	362956003	0	387713003
309902039	362956003	1	108290001
309902039	362956003	2	6996004

4.4.2.5.2 Displaying a navigation subset

Displaying a *navigation subset* is very similar to *using | Is a | relationships for hierarchy navigation*. Having mastered the techniques for using the | is a | *relationship*, the key to using *Navigational Hierarchies* lies in seeing which columns of the *Subset Members* table correspond to similar columns in the *Relationship* table. This is illustrated in *Table 251*.

Table 251: Hierarchy Representation in Relationships and Navigation Subsets

Usage	***Relationship*** **Field**	Navigation SubsetMember Field
Identify the hierarchy	RelationshipType	*SubsetId*
Identify the parent *concept*	*ConceptId2*	*MemberId*
Identify the child *concept*	*ConceptId1*	*LinkedId*
Specify order of child under the parent	(not applicable)	*MemberStatus*

The example *subset members* shown in *Table 253* arrange the *concepts* listed in *Table 2* into the *navigation hierarchy* shown in. *Figure 49*.

Table 252: Concepts used in the Example Navigation Subset Set

Id	*Preferred Term*
1225002	| radiography of humerus |
1597004	| skeletal X-ray of ankle and foot |
168594001	| clavicle X-ray |
168619004	| plain X-ray head of humerus |
168620005	| plain X-ray shaft of humerus |
168623007	| X-ray shaft of radius/ulna |
168637003	| plain X-ray radius |
168655007	| instability views carpus |

Id	*Preferred Term*
168663008	| plain X-ray head of femur |
168664002	| femoral neck X-ray |
168665001	| plain X-ray shaft of femur |
168669007	| patella X-ray |
205115004	| radiologic examination of femur, anteroposterior and lateral views |
241063007	| bicipital groove X-ray |
241066004	| ulna groove X-ray |
241069006	| ulna X-ray |
241071006	| scaphoid X-ray |
241073009	| metacarpal X-ray |
241075002	| femur X-ray |
241076001	| tibia and/or fibula X-ray |
241077005	| tibia X-ray |
241078000	| fibula X-ray |
241079008	| metatarsal X-ray |
241080006	| tarsus X-ray |
268427003	| X-ray shaft of tibia/fibula |
271311001	| carpal bones X-ray |
302402006	| radius and/or ulna X-ray |
37815002	| diagnostic radiography of calcaneus |
40348008	| skeletal X-ray of pelvis and hip |
418687005	| fluoroscopy of humerus |

Id	*Preferred Term*
427961005	| x-ray of acetabulum |
432552002	| computed tomography of clavicle |
48966008	| skeletal X-ray of shoulder and upper limb |
5433008	| skeletal X-ray of lower limb |
70780000	| skeletal X-ray of elbow and forearm |
72872009	| skeletal X-ray of upper limb |
79082005	| diagnostic radiography of fibula, combined AP and lateral |
82420003	| radiologic examination of forearm, anteroposterior and lateral views |

Table 253: Navigation Subset Member Example

SubsetId	*MemberId*	*MemberStatus*	*LinkedId*
<*SubsetId*>	<NavConceptId-A>	1	5433008
<*SubsetId*>	5433008	1	241080006
<*SubsetId*>	241080006	1	37815002
<*SubsetId*>	5433008	2	241079008
<*SubsetId*>	5433008	3	241076001
<*SubsetId*>	241076001	1	241078000
<*SubsetId*>	241078000	1	268427003
<*SubsetId*>	241078000	2	79082005
<*SubsetId*>	241078000	3	241077005
<*SubsetId*>	5433008	4	241075002
<*SubsetId*>	241075002	1	205115004
<*SubsetId*>	241075002	2	168665001

SubsetId	***MemberId***	***MemberStatus***	***LinkedId***
<SubsetId>	241075002	3	168664002
<SubsetId>	241075002	4	168663008
<SubsetId>	5433008	5	168669007
<SubsetId>	5433008	6	1597004
<SubsetId>	5433008	7	40348008
<SubsetId>	40348008	1	427961005
<SubsetId>	<NavConceptId-A>	2	72872009
<SubsetId>	72872009	1	302402006
<SubsetId>	302402006	1	241069006
<SubsetId>	241069006	1	168623007
<SubsetId>	302402006	2	168637003
<SubsetId>	302402006	3	70780000
<SubsetId>	70780000	1	241066004
<SubsetId>	302402006	4	82420003
<SubsetId>	72872009	2	168594001
<SubsetId>	168594001	1	432552002
<SubsetId>	72872009	3	1225002
<SubsetId>	1225002	1	241063007
<SubsetId>	1225002	2	168620005
<SubsetId>	1225002	3	168619004
<SubsetId>	1225002	4	418687005
<SubsetId>	72872009	4	168655007
<SubsetId>	72872009	5	271311001

SubsetId	***MemberId***	***MemberStatus***	***LinkedId***
<SubsetId>	271311001	1	241071006
<SubsetId>	72872009	6	241073009
<SubsetId>	72872009	7	48966008

- 5433008 | skeletal X-ray of lower limb |
    - 241080006 | tarsus X-ray |
        - 37815002 | diagnostic radiography of calcaneus |
    - 241079008 | metatarsal X-ray |
    - 241076001 | tibia and/or fibula X-ray |
        - 241078000 | fibula X-ray |
            - 268427003 | X-ray shaft of tibia/fibula |
            - 79082005 | diagnostic radiography of fibula, combined AP and lateral |
            - 241077005 | tibia X-ray |
    - 241075002 | femur X-ray |
        - 205115004 | radiologic examination of femur, anteroposterior and lateral views |
        - 168665001 | plain X-ray shaft of femur |
        - 168664002 | femoral neck X-ray |
        - 168663008 | plain X-ray head of femur |
    - 168669007 | patella X-ray |
    - 1597004 | skeletal X-ray of ankle and foot |
    - 40348008 | skeletal X-ray of pelvis and hip |
        - 427961005 | x-ray of acetabulum |
- 72872009 | skeletal X-ray of upper limb |
    - 302402006 | radius and/or ulna X-ray |
        - 241069006 | ulna X-ray |
            - 168623007 | X-ray shaft of radius/ulna |
        - 168637003 | plain X-ray radius |
        - 70780000 | skeletal X-ray of elbow and forearm |
            - 241066004 | ulna groove X-ray |
        - 82420003 | radiologic examination of forearm, anteroposterior and lateral views |
    - 168594001 | clavicle X-ray |
        - 432552002 | computed tomography of clavicle |
    - 1225002 | radiography of humerus |
        - 241063007 | bicipital groove X-ray |
        - 168620005 | plain X-ray shaft of humerus |

    - 168619004 | plain X-ray head of humerus |
    - 418687005 | fluoroscopy of humerus |
- 168655007 | instability views carpus |
- 271311001 | carpal bones X-ray |
    - 241071006 | scaphoid X-ray |
- 241073009 | metacarpal X-ray |
- 48966008 | skeletal X-ray of shoulder and upper limb |

Figure 49: Example Navigation Subset - Hierarchy View

4.4.2.5.3 Completeness of navigation hierarchies

There is no particular requirement for navigational *Subsets* to contain links that will allow access to all *descendants* of a *Concept* included in the navigational *Subset*. Implementers of *SNOMED CT* may choose to display only those *Concepts* they have included in the *navigation Subset Members* table. Alternatively they may wish to use the *navigation Subset* to organize the top-level *hierarchy*, but then switch to using the | is a | *hierarchy* from the *Relationship* table when *navigation* reaches a certain depth.

4.4.2.6 Using Tree View Components for Hierarchy Display

The two examples given below show the creation of a tree view from a small sample *hierarchy*.

The principals used can be extended to any size or depth of *hierarchy*.

4.4.2.6.1 Example 1: Show all descendants of Concept "A" in a tree view

Table 254: Relationships used in the Example

ConceptId1	*Relationship*	*ConceptId2*
B	| is a |	A
C	| is a |	A
D	| is a |	B
E	| is a |	B
E	| is a |	C
C	| is a |	F

We must process each *concept* in the *hierarchy*, starting at 'A'. Add a tree node for 'A', and then *query* to get the *children* of 'A'. Process each *child* recursively, i.e. add a node to the tree view for the *child*, then *query* for its *children*, etc.

Table 255: Concept to node cross reference

Node	*Child* Node
1	2

Node	***Child* Node**
1	5
2	3
2	4
5	6
5	7

Table 256: Child nodes

Node	***ConceptId***
1	A
2	B
3	D
4	E
5	C
6	E
7	F

Now we have tree nodes and their *children* for each *Concept*. If the nodes have been added to a Windows tree view component, display will be automatic. If a text-based display is being used then the nodes can be output to the screen using the indent style display. Note that the *Concept* 'E' appears in the tree view twice, under each of its parents.

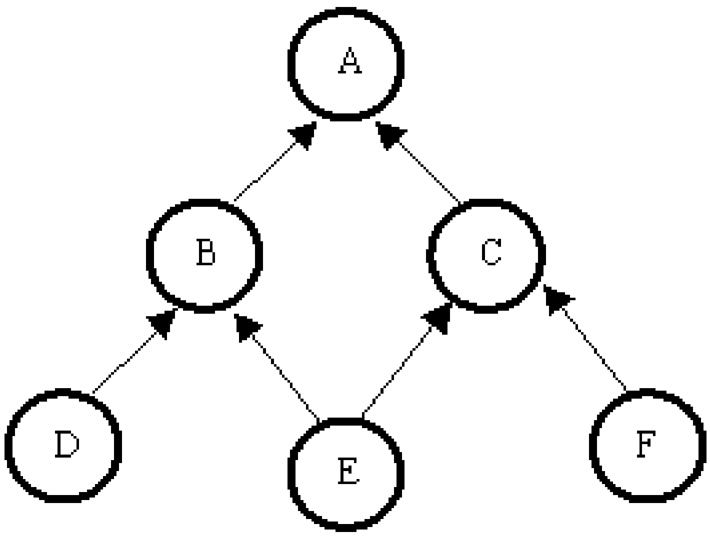


Figure 50: Graphical view of sample hierarchy - descendants of "A"

- (1) **A**
  - (2) **B**
    - (3) **D**
    - (4) **E**
  - (5) **C**
    - (6) **E**
    - (7) **F**

Figure 51: Tree view of sample hierarchy

4.4.2.6.2 Example 2 - Show all ancestors of Concept "E"

In order to construct the tree view, we must start from the top down, so we must create a temporary view of the *hierarchy* before we can add nodes to the tree view. *Query* to get the parents of 'E'. Process each parent recursively, i.e. add an entry to the temporary table, stating that 'E' is a *child* of each of its parents, then *query* to get its parent, etc. When the top of the tree is reached, a record is kept of the top-level *Concept*, since this will be the starting point for building the tree view.

Table 257: Temporary view of the hierarchy

Concept	***Child Concept***
B	E
C	E
A	B
A	C

We can now use the temporary table information to build the tree view from the top down. Starting at A, add a node to the tree view. Work recursively from the information in the temporary view of the *hierarchy* to add the *descendants* of 'A' into the tree view.

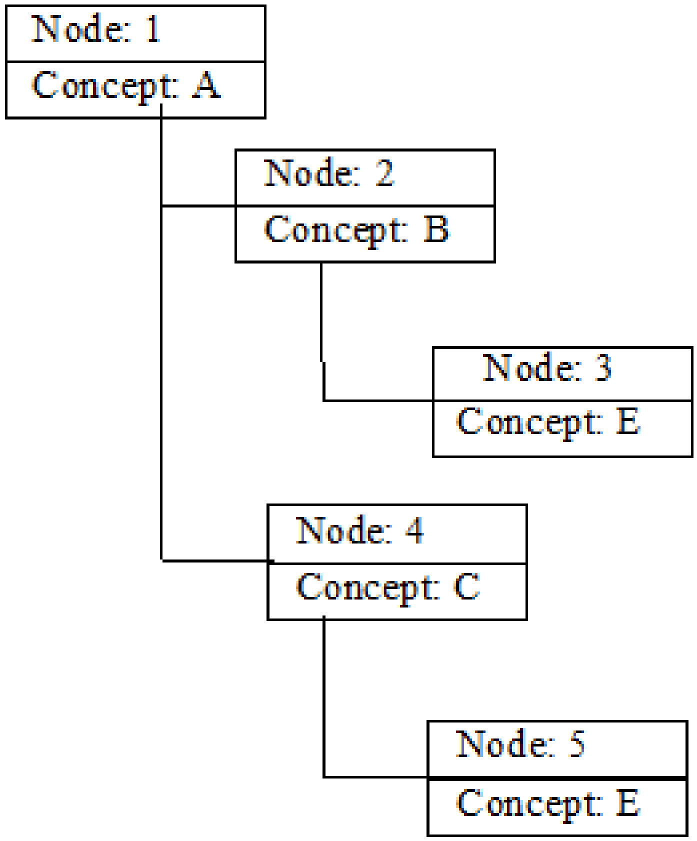


Figure 52: Tree view of sample hierarchy - ancestors of "E"

4.4.3 Applying Subsets

A *Subset* refers to a set of *Concepts*, *Descriptions*, or *Relationships* that are appropriate to a particular *language*, *dialect*, country, specialty, organization , user or context.

In its simplest form, a *Subset* is a list of *SNOMED Clinical Terms Identifiers* (*SCTIDs*). Each *SCTID* refers to one component of *SNOMED CT*, and makes it a member of the *Subset* (a *Subset Member*).

Example:

As an analogy, think of *SNOMED CT* is a book. A *Subset* is like an index entry pointing to a set of pages relevant to a particular topic.

Subsets may be used to derive tables that contain only part of *SNOMED CT*. In some cases, these cut-down "Standalone *subset*" tables may also be centrally distributed (e.g. a release table containing only *Descriptions* that are valid in a chosen *language*). The *Subsets* described in this guide are additional tools for enhancing the usability of *SNOMED CT*.

Complete technical information on *subsets*, including the *subset* table design can be found in the *Subset Mechanism* specification.

4.4.3.1 Subset Types

SNOMED CT supports the *Subset Types* listed in the following table.

Table 258: Summary of Subset Types

Subset Type	**Summary**
Language Subset	The *Descriptions* applicable to a *Language* or *Dialect*. May indicate whether each *Description* contains the *Preferred Term*, *Synonym*, or *Fully Specified Name* for the associated *Concept* in that *language* or *dialect*.
Realm Concept Subset	The *Concepts* applicable for a particular *Realm*. May assign priorities to *Concepts* appropriate to the *Realm*.
Realm Description Subset	The *Descriptions* applicable for a particular *Realm*. May indicate whether each *Description* contains the *Preferred Term* or a *Synonym* used in that *Realm*.
Realm Relationship Subset (for future use)	The *Relationships* applicable to a particular *Realm*.
Context Concept Subset	The *Concepts* applicable to a particular *Context Domain*.
Context Description Subset	The *Descriptions* applicable to a particular *Context Domain*.
Navigation Subset	A set of *Navigation Links* representing an ordered *hierarchy* appropriate for display and user *navigation*.
Duplicate Terms Subset	Indicates the suggested priority among sets of *Descriptions* containing identical *terms*.

4.4.3.2 Subset Members

The format of the *Subset Members Table* appears quite simple, containing only 4 fields:

- *SubsetId*
- MemberID;
- *MemberStatus*
- LinkedID.

There are some underlying subtleties to this table however.

The LinkedID field is only valid for:

- *Navigation Subsets*, where it specifies the navigational *children* of the *subset member*;
- *Duplicate Terms Subsets*, where it refers to the *Description* which is most highly prioritized.

The field *MemberStatus* has a different significance depending on the *type of subset* specified in the *Subset* Table.

A brief description of the significance of the *MemberStatus* in each *type of subset* is given below.

4.4.3.2.1 Significance of MemberStatus in a Language or Realm Description Subset

The *MemberStatus* specifies the role of the reference *Description* in this *Language* or *Realm*. There are three roles.

1 = *Preferred Term*

2 = *Synonym*

3 = *Fully Specified Name*

Note that the following rules apply to each *Subset*:

- A *Description* cannot be assigned the role *Fully Specified Name* unless its *DescriptionType* is also "*Fully Specified Name*.";
- Only one *Description* of each *Concept* may be assigned as the *Fully Specified Name*;
- If no *Description* of a *Concept* is assigned the role *Fully Specified Name*, the FullySpecifiedName in the *Concepts Table* assumes this role;
- A *Description* cannot be assigned the role *Preferred Term* or *Synonym* if its *DescriptionType* is "*Fully Specified Name*.";
- One and only one *Description* of each *Concept* must be assigned the role "*Preferred Term*.";
- Any number of *Descriptions* of a *Concept* may be assigned the role "*Synonym*.".

4.4.3.2.2 Significance of MemberStatus in a Realm Concept or Context Concept Subset

The MemberStatusspecifies the priority of the *Concept* within the *Subset*, with lower numbers denoting higher priority. The priority may be used to determine *order* of display. *Concepts* with priority *status* of "0" would appear at the top of the list.

4.4.3.2.3 Significance of a MemberStatus in a navigational Subset

The *MemberStatus* specifies the *order* of the *child concepts* within the navigational *hierarchy*. This may be used to determine the display *order* of *children* when displaying a navigational *Subset*.

4.4.3.2.4 Significance of MemberStatus in a Realm Relationship (for future use) or Context Description Subset

The *MemberStatus* has no particular significance in these types of *subsets*, and will be set to the value of "1,"meaning include this table row in the *subset*.

4.4.3.3 Using subset definition files (future use)

Subset definition files are the preferred way of specifying the content of *subsets*. The *subset definition file* is a series of logical clauses, expressed as XML mark-up, which defines what should be included in the *subset*. While all *subsets* released by the *IHTSDO* are currently distributed in the relational file format, there are plans to provide these *subsets* as XML definition files. This method of specifying *subsets* has several advantages:

- It is less verbose as many members can be included by a single clause.
- It is likely to be resilient to the effects of applying updates and *Extensions*.
- One *Subset* can be used as the basis for another *Subset*. The defining clauses express differences between the base *Subset* and the defined *Subset*.

Example:

A UK English *dialect Subset* may be expressed as differences from an International English *language Subset.*

- *Members* can be added to a *Subset* according to hierarchical *Relationships*.

Example:

A single defining clause can add all | is a | *descendants* of the *Concept* "Musculoskeletal procedure" to an orthopedic specialist *Subset.*

Software developers who wish to use *subset definition files* will need to produce software that will generate relational *subset* tables from a *subset definition*. Please contact the *IHTSDO* for *current* information as this format evolves.

4.4.3.4 Using relational Subset tables

Any *Subset Definition* can be automatically converted into the relational form. However, the relational form is *release version* dependent whereas some *Subset Definitions* may be applicable to more than one release of *SNOMED CT*.

An application may use the relational form of a *Subset* in one of three ways:

1. As part of a run - time *query* to retrieve, filter or sort members of a *Subset*.

   Example:

   An SQL *query* with a join between the *Subsets Table* and a *core table* could be used to:

   - Provide a pick list of all members of a *Subset*;
   - Filter a search or hierarchical list so that only members of the *Subset* are shown;
   - *Order* a search or hierarchical list according to the sequence specified in the *Subset*.

2. To pre-filter a *SNOMED CT* table when it is imported from a *release file*.

   Example:

   You might choose to exclude *Descriptions* in a *language* or *dialect* not required.

   - Care should be taken when considering the use of pre-filtering with *Concepts*. If *SNOMED CT* data from other systems is received it will not be possible to interpret it unless the relevant *Concepts* (including *Concepts* used directly or indirectly in a definition) are present on the system.

3. To modify the application's internal representation of *Concepts* or *Descriptions* to allow efficient application of the *Subset*.

   - Possible modifications include addition of flags or indexes or some type of pre-sorting of data.
   - Objectives of this include:
     - Optimizing frequently used *Subsets*
     - Facilitating rapid switching between different *Subsets* depending on context or user configuration.

Note: The *release file* formats are tab-delimited files that allow the information to be imported to any relational database management system.

4.4.3.5 Practical uses of Subsets

Subsets can be used for many different purposes. This section outlines some of the ways in which the mechanism specified in this guide can be used to meet different practical requirements. The uses outlined are illustrative examples and do not represent all possible applications of *Subsets*.

4.4.3.5.1 Controlling data entry

Many clinical applications include facilities for data entry to be controlled or assisted by protocols, templates or structured data entry forms. Different sets of candidate *Terms* or *Concepts* may be appropriate to each data entry field. The sets of candidate *Terms* or *Concepts* for a field may be very large (e.g. any operative procedure) or very small (e.g. the possible observations from a particular examination).

The *Subset* mechanism allows *Concepts* relevant to particular uses to be specified as follows:

- A *Context Concept Subset* or *Context Description Subset* can be specified for each data entry field.
- A *Context Concept Subset* refers to all the *Concepts* that are applicable to a particular category of use:
  - A *Context Concept Subset* allows any *Description* associated with the set of permitted *Concepts* to be used (i.e. a *Preferred Term* or a *Synonym*).
- *Context Description Subset* refers to all the *Descriptions* that are applicable to a particular field:
  - A *Context Description Subset* only allows a specified set of *Descriptions* to be used. The permitted set of *Concepts* is implicitly any *Concept* associated with permitted *Description*.

A *SNOMED CT enabled application* can make use of Context *Subsets* if it is able to associate data entry fields with an identifiable *Context Domain*. There are several alternative ways of using Context *Subsets*. These include:

- For fields associated with a Context *Subset* containing only a small number of permitted values, the members of the *Subset* may be used to populate a drop-down list or another similar option selection tool;
- For fields associated with a larger range of options, a Context *Subset* may be used as a filter for searches;
- When using automated text encoding tools a Context *Subset* may limit the scope of possible encoding and thus should accelerate and improve the accuracy of parsing.

4.4.3.5.2 Using terminology in electronic health records

When a user searches *SNOMED CT*, they should only see *Descriptions* associated with *Concepts* that are usable in the context in which they are working. The *Concepts* that form part of *SNOMED CT* have various different uses. There are many examples of *Concepts* that are meaningless if used in an inappropriate context. These include:

- *Concepts* that are used purely to structure and organize the terminology but which would be meaningless in a patient record (e.g. the root *Concept*, top-level *Concepts*).
- *Concepts* that may be useful as headings for sections in a record but which convey no clinical meaning (e.g. general *Concepts* like "Laboratory results," "Medication").
- *Concepts* that refer to an organism or material which are only meaningful in the context of other information (e.g. "Aspirin" is meaningful in a patient record only as part of, or in relation to, another *Concept* such as "Overdose," | Allergic reaction | or "Prescribed").
- *Concepts* that refer to disorders are meaningless or misleading if used to record a procedure.

The *Subset* mechanism allows *Concepts* relevant to particular uses to be specified as follows:

- A *Context Concept Subset* can be specified for any distinct category of use associated with a set of *Concepts*.
- A *Context Concept Subset* contains a RealmID and ContextID which identify the *Context Domain* to which it applies.
- A *Context Concept Subset* refers to all the *Concepts* that are applicable to a particular category of use.
- A *Context Concept Subset* only refers to *Concepts*. However, it also affects the availability of *Descriptions*. A *Description* associated with a *Concept* that is not used in a particular context is not accessible.

A *SNOMED CT enabled application* should be able to recognize usage categories that are relevant to it and should apply the appropriate *Context Concept Subsets* to:

- Restrict access to *Concepts* (and associated *Descriptions*) so that only those referred to by the *Context Concept Subsets* are accessible.

4.4.3.5.3 Languages and Dialects

SNOMED CT is designed as a multilingual terminology.

The *Subset* mechanism enables *SNOMED CT* to support different *Languages* and *Dialects* in the following way:

- There is a *Language Subset* for each supported *Language* and *Dialect*.
- The *LanguageCode* identifies the *Language* or *Dialect*.
- A *Language Subset* refers to all the *Descriptions* that contain *Terms* expressed in that *Language* or *Dialect*.
- Each member of a *Language Subset* has a *MemberStatus*. This indicates the use of the *Description* in relation to its associated *Concept*. In each *Language* or *Dialect*, a *Description* may represent a:
  - *Preferred Term*
  - *Synonym*
  - *Language* -specific *Fully Specified Name*

A *SNOMED CT enabled application* should be able to:

- Allow a selection of a particular *Language Subset* as a configuration option;
- Restrict access to *Descriptions* so that only those referred to by a selected *Language Subset* are accessible;
- Treat the *Preferred Terms* and *language* -specific *Fully Specified Name* referred to by a selected *Language Subset* in a manner appropriate to their specified usage.

Some applications may also support multiple *Languages* or *Dialects* allowing selection of combinations of *Language Subset*.

4.4.3.5.4 Managing duplicate terms

A *Fully Specified Name* in a given *Language* or *Dialect* is uniquely associated with one and only one *Active Concept*. However, this requirement does not apply to other *Terms* (i.e. *Preferred Terms* and *Synonyms*).

SNOMED CT permits the same *Term* to occur in more than one *Active Descriptions* each associated a different *Active Concept*. This enables *Preferred Terms* and *Synonyms* to be phrased in the natural *languages* of clinicians without discriminating between those in different disciplines or specialties who use the same words or phrases to mean different things.

Example:

- "Liver" - may refer to a foodstuff (dietitian) or to a body organ (surgeon);
- "Fundus" - may refer to part of the eye, stomach or uterus depending on specialty and context;
- "Aspiration of stomach content" - may refer to a procedure (following poisoning) or a complication of anesthesia or childbirth;
- "Hypertension" - may refer to a finding (raised blood pressure) or to a recognized diagnosis with particular diagnostic criteria.

Similarly the same words or phrases may be used to refer to the *Concept* with different levels of precision.

Example:

| Arm | - may refer to the entire arm (as in "amputation of arm") or to part of the arm (as in "injury to the arm").

The existence of multiple *Descriptions* containing exactly the same *Term* but linked to different *Concepts* does not cause any technical problems but may cause confusion at the *user interface*. The appearance of several identical phrases in a search list presents a dilemma and applications need to present sufficient information to allow the user to choose between them. In most cases this requires the application to provide a view of the *Fully Specified Names* associated with any *Duplicate Terms*.

The *Duplicate Terms Subset*, part of the *SNOMED CT* Developer Toolkit, is intended to assist applications to identify and manage the presentation of *Duplicate Terms* by:

- Identifying the *Duplicate Terms* in a particular release of *SNOMED CT*.

This simplifies the processing required to determine if a duplicate exists and will require disambiguation.

- Indicating a priority between *Duplicate Terms*.

Applications may use these priorities to support sensible display ordering or default selection.

The *Duplicate Terms Subset* is *Language* and *Dialect* dependent since the occurrence of and priorities between duplicates will vary according to the applicable set of *Terms* specified in a *Language Subset*. In addition, specialty or user specific priorities may be applied by alternative *Duplicate Terms Subsets* derived from the distributed files.

- Each *current Description* containing a *Term* that is also present in another *current Description* is a member of the *Duplicate Terms Subset*;
- The *Subset Member* has a *status* that indicates its priority;
- The LinkedID refers to the *Description* which contains the same *Term* and has the highest priority.

This reference links together a set of *Descriptions* which contain the same *Term*.

4.4.3.5.5 Specifying "natural" display orders

The formal *subtype Relationships* of *SNOMED CT* are specified as logical definitions. These provide a powerful tool for organizing , retrieving and aggregating data. These *subtype Relationships* can also be used for *Navigation* to locate a more or less specific *Concept*. However, the requirements for *Navigation* differ from those for logical definition. The following factors affect the efficiency of *Navigation* but have no relevance to logical definitions of *Concepts*:

- Frequency of use of *Concepts* and their relative accessibility.
- Depth of nesting of a *hierarchy*
- Length of a displayed *hierarchy* level.
- *Order* of listed *Concepts*:
    - Matching a natural *order* (e.g. cranial nerves, cervical vertebrae, etc.).
    - Reflecting frequency of use.
- User association of ideas that are not linked as *subtypes* of one another.

The *Relationships Table* does not specify the *order* in which *Relationships* should be displayed and places no limits on the size or depth of any *hierarchy*. There are several *reasons* for this:

- The display *order* of many *Relationships* is arbitrary.
- Different display *orders* may be appropriate to the same *Relationships* depending on user preferences and the context.
- The display *order* of *Relationships* is not fundamental to an understanding of the related *concepts*. A change in display *order* does not alter the meaning of a *Concept*. Therefore, it need not be subject to the rigorous change control applied to the *core tables*.

There is a requirement for one or more *Navigation* hierarchies, which are configurable without undermining the logical definitions of *SNOMED CT*.

The *Subset* mechanism includes *Navigation* hierarchies, which allow a natural or preferred display *order* to be defined for a set of *Navigation Links*. They are represented as follows:

- A *Navigation Subset* specifies an ordered list of *Navigation Links* for *Concepts* that act as nodes in the *hierarchy*.
- Each *Navigation Link* is represented as a member of the *Navigation Subset*
- Each *Navigation Link* specifies the *ConceptIds* of a parent *Concept* and a *Navigation child Concept* and an *integer* used to *order* the list of *children*.
- A *Navigation hierarchy* may include both ordinary *SNOMED CT Concepts* and special *Navigation Concepts*.
- *Navigation Concepts* exist only for the purpose of *Navigation* and are not suitable for recording or aggregating information.
- This allows a *Navigation hierarchy* to contain nodes which represent user recognizable composite *Concepts* that are not part of the formally defined content of *SNOMED CT*.
- A *Navigation hierarchy* may be similar to the *subtype hierarchy* specified by | is a | *Relationships* but it differs in the following ways:

- *Navigation Links* need not represent logical semantic *relationships* between *Concepts*. They can be used to link to different *Concepts* that a user may seek in an indirectly associated part of the *SNOMED CT hierarchy*.
- *Navigation Links* need not include a complete set of *subtypes* for a *Concept*. Instead they may be limited to those most commonly used.
- *Navigation Links* can skip levels in the *subtype hierarchy* to show commonly used *Concepts* to a higher more readily accessible level.
- *Navigation Links* can use *Navigation Concepts* to add intermediate levels into the formal *subtype hierarchy* to rationalize screen displays.
- *Navigation Links* are ordered and thus provide added value even when they have a one-to-one *relationships* with the *subtype hierarchy* (for example, while the |is a| *Relationships* identify all the cranial nerves a set of *Navigation Links* can *order* them in the expected *order* 1 to 12).

A *SNOMED CT enabled application* should support ordered hierarchical displays based on the *Navigation Links* in a selected *Navigation Subset*. *SNOMED CT enabled applications* should also provide access to the *subtype hierarchy* so that a complete set of *subtypes* can be accessed where required. A *Navigation* display should therefore always include an option to switch to a *subtype* view of the same focus *Concept*.

4.4.3.5.6 National requirements for specific Concepts

SNOMED CT is designed for use in many different countries and consequently includes a certain number of country-specific *Concepts*. A country, such as the United Kingdom for example, may have specific requirements for the representation of *Concepts* that are not meaningful in the United States or other countries. These variations are particularly significant for the interfaces between clinical care, service administration and reimbursement. National laws and conventions may also create additional *refinements* of more general *Concepts*.

A user or group of users usually only requires access to *Concepts* deemed to be relevant in the country in which they are working.

The *Subset* mechanism meets the requirement for national and regional variations in the use of *Concepts* in the following ways:

- *Realm Concept Subset* can be specified for each country in which *SNOMED CT* is used.
- If a country has administrative regions with different requirements, these can also be specified with a *Realm Concept Subset*.
- A RealmID identifies the country or region.
- A *Realm Concept Subset* refers to all the *Concepts* used in that country or administrative region.
- A *Realm Concept Subset* only refers to *Concepts*. However, it also affects the availability of *Descriptions* and *Relationships*. A component that refers to a *Concept* that is not used in a *Realm* is implicitly also excluded from use in that *Realm*.

A *SNOMED CT enabled application* should be able to:

- Restrict access to *Concepts* so that only those referred to by a selected *Realm Concept Subset* are accessible;
- Restrict access to *Descriptions* so that only those associated with *Concepts* referred to by a selected *Realm Concept Subset* are accessible;
- Restrict use of *Relationships* so that any *Relationship* that refers to a *Concept* that is not referred to by the *Realm Concept Subset* is rendered unusable.

Example:

- An oncology subspecialty organization *Realm Concept Subset* may refer to the hierarchical *descendants* of a general *Concept* representing "Grant funding for cancer research";
- A *UK NHS Realm Concept Subset* may refer to the hierarchical *descendants* of a general *Concept* representing " *NHS* National Service Frameworks.".

4.4.3.5.7 Regional variations in disease prevalence

There are substantial differences in the prevalence of diseases in different regions in which *SNOMED CT* may be used. Users will expect to find the conditions they commonly deal with, without being distracted by long lists of conditions they rarely see.

In the main body of *SNOMED CT*, all *Concepts* are equally visible irrespective of their relative prevalence. The *Subset* mechanism can be used for prioritizing the *Concepts* in the following way:

- *Realm Concept Subset* can be specified for a region with a particular pattern of disease prevalence.
- A RealmID identifies the region.
- A *Realm Concept Subset* refers to all the *Concepts* used in that region. For each of these *Concepts*, the *Subset* may specify a *MemberStatus* that assigns a priority to that *Concept*. *Concepts* that are used most frequently in a region are assigned first priority (*MemberStatus* =1). There are nine priority levels (*MemberStatus* =1 to 9). The least frequently used *Concepts* are assigned the lowest priority (*MemberStatus* =9).
- A *Realm Concept Subset* only refers to *Concepts*. However, the assigned priority also affects the *Descriptions* associated with those *Concepts*.

A *SNOMED CT enabled application* should be able to:

- Prioritize access to *Concepts* relative to one another according to the *MemberStatus* specified by a selected *Realm Concept Subset*;
- Prioritize access to *Descriptions* relative to one another according to the *MemberStatus* assigned to the associated *Concepts* by a selected *Realm Concept Subset*.

The way in which access is prioritized depends on the nature of the application and its operating environment. However, examples of prioritization include:

- Showing *Descriptions* associated with high priority *Concepts* before those with lower priority when searching for word or phrases;
- Showing *Concepts* with high priority before their less highly prioritized siblings in hierarchical displays;
- Initially listing *Concepts* and associated *Descriptions* with priority above a specified threshold and requiring an additional step to access those assigned lower priority.

4.4.3.5.8 Specialty and discipline-dependent variations in use of Concepts

SNOMED CT contains *Concepts* used by many different groups of health professionals. The frequency of use of these *Concepts* depends on the professional discipline and/or clinical specialty of the user. It is important to ensure that the user is able to access the *Concepts* that they use frequently, without being distracted by thousands of textually similar *Concepts* they rarely require.

The *Subset* mechanism can be used for prioritizing the *Concepts* in the following way:

- A *Realm Concept Subset* can be specified for each clinical specialty and/or discipline.
- A RealmID identifies the specialty.
- A *Realm Concept Subset* refers to all the *Concepts* used in that specialty. For each of these *Concepts*, the *Subset* may specify a MemberStatusthat assigns a priority to that *Concept*. There are nine priority levels from 1 (highest priority) to 9 (lowest priority).
- A *Realm Concept Subset* only refers to *Concepts*. However, the assigned priority also affects the *Descriptions* associated with those *Concepts*.

4.4.3.5.9 Local needs of organizations or individual users

The previous sections have dealt with requirements of countries, regions and specialties. Organizations and individual users may also have similar requirements for restricting or prioritizing access to particular *Concepts*.

The *Realm Concept Subsets* described in the previous sections may also be applied at a more local level to meet the needs of an organization , user or group of users.

In practice , a localorganization or individual user will often need a *Realm Concept Subset* derived by merging or refining other *Realm Concept Subsets*.

Example:

A cardiologist, working in "Any Town General Hospital," an *NHS* hospital in the UK, may be best served by a *Subset* derived from a combination of the existing *Realm Concept Subsets* specified for:

- *UK NHS*
- Cardiology specialty.
- Any Town General Hospital.
- With modifications for any local factors (e.g. special research interests of the local cardiology department or clinician).

4.4.3.5.10 National, organizational and specialty variations in term preferences

In addition to requirements for different *Concepts*, a country, organization or specialty may have different preferences for the *Terms* to be used to describe identical *Concepts*.

The *Subset* mechanism meets the requirement for national and region variations in the use of *Terms* in the following way:

- Any *Realm* that requires a specialized *refinement* of a *Language* or *Dialect* may specify a *Realm Description Subset*.
- A *Realm Description Subset* is similar to a *Language Subset* but differs in two ways:
  - It contains a RealmID that identifies a *Realm* within which it applies;
  - It must not be used to override the *language* -specific *Fully Specified Name* role assignments made by a *Language Subset*.
- Support for *Term* variations is *Language* or *Dialect* dependent. Therefore, in a multilingual country or organization , there may be a single *Realm Concept Subset* but several separate *Realm Description Subsets*.

A *SNOMED CT enabled application* should be able to:

- Use a selected *Realm Description Subset*, in place of a *Language Subset*, to restrict access to *terms* and to assign the *Preferred Term* and *Synonym* roles to different *Descriptions*.

4.4.3.5.11 Managing the coded content of messages

Healthcare messages include fields that can be populated with codes from clinical coding schemes. *SNOMED CT* provides *concept identifiers* as a means of encoding *Concepts*. These *concept identifiers* are suitable for use in appropriate fields of many clinical messages.

Implementations of clinical messaging typically constrain the range of values that can be applied to particular fields. There are several *reasons* for this:

- To ensure that the information encoded is meaningful as a value for the specified field.

Example:

A field that is intended to describe the nature of investigation may contain a code that means "Serum glucose measurement" but should not contain a code that means "Hypoglycemia."

- To ensure that receiving application is able to process the message.

Example:

A locally added code value may be valid in a particular application but should not be used if the receiving application needs to retrieve, process or analyze the coded part of the message.

- To ensure adequate detail and specificity.

Example:

A field used to report an operative procedure could contain a code for "Abdominal procedure." However, this would not be adequate to meet the business purpose served by a message.

- To avoid unnecessary detail or diversity.

Example:

A biochemical investigation could be reported using a code that represents various detailed aspects of the method used to perform the investigation. Such details may be unnecessary to a clinician and may complicate the analysis, charting and graphing of a series of results reported at different levels of detail.

Practical examples of these *constraints* are:

- The *UK NHS* "bounded-lists" of codes that are permitted for use in specified fields of the pathology report message;
- The *HL7* specification of "Vocabulary *Domains*" associated with every coded field in *HL7* Version 3 messages;
- The members of the *SNOMED CT* US Drug *Extension* that contains pharmaceuticals approved for distribution in the United States.

The *Subset* mechanism allows Vocabulary *Domains* (or bounded-lists) of *SNOMED CT Concepts* to be specified in the following way:

- A *Context Concept Subset* is specified for each Vocabulary *Domain*;
- A *Context Concept Subset* lists the *SCTIDs* of all *Concepts* applicable to the Vocabulary *Domain* associated with a particular message field.

Where appropriate, a *SNOMED CT enabled application* should be able to use *Context Concept Subsets*:

- To generate messages that contain *SCTIDs* representing *SNOMED CT Concepts* appropriate to particular message fields;
- To validate messages to ensure that no fields contain *SCTIDs* that are inappropriate to the relevant Vocabulary *Domain*.

4.4.4 Access to qualifiers and refinable characteristics

A *terminology server* should enable an application to review the refinable *defining characteristics* and the specified set of *qualifying characteristics* for any selected *Concept*.

4.5 Testing and traversing subtype Relationships

The *subtype hierarchy* represented by | is a | *relationships* is an essential element in the structure and semantics of *SNOMED CT*. All *SNOMED CT enabled terminology servers* need to provide functions that test and traverse these *relationships* to navigate the hierarchy and to determine whether a *concept* is a *subtype* of another specified *concept*.

4.5.1 Top-level ancestor checking

Terminology servers should allow client applications to rapidly determine the top-level *Concept* that is the *supertype ancestor* for any specified *Concept*.

Each *Concept* has only one top-level supertype and this represents the semantic-type of the *Concept*.

4.5.2 Navigation concept checking

Terminology servers allow client applications to determine whether a specified *Concept* is a *navigation Concept*.

4.5.3 Subtype descendant testing

Terminology servers should be able to test whether any specified *Concept* is a *descendant subtype* of another specified *Concept*.

4.5.4 Subtype search scope restriction

Terminology servers should be able to restrict searches so that they return only those *Concepts* and (or their associated *Descriptions*) that are *subtype descendants* of a specified *Concept*.

Subtype search scope restriction is particularly valuable with respect to top-level *Concepts*. For example, when searching for a procedure it is useful to be able to exclude disorders or findings that may contain similar words of phrases.

Generalizing *subtype* search scope restriction to other nodes in the *subtype hierarchy* may significantly enhance usability in some situations.

Example:

When undertaking an ophthalmologic examination, a search for findings could be constrained to findings related to the eye, increasing the specificity of results of searches for phrases containing the word "fundus."

4.5.5 Optimizing concept subsumption testing

Rapid and efficient computation of whether a *concept* is a *subtype descendant* of another *concept* is essential for effective *transformation* of *expressions* and for testing subsumption between *expressions*.

4.5.5.1 Approaches to concept subsumption testing

There are several strategies for delivering efficient computation of subsumption between *concepts*. These are summarized in the following subsection with a brief evaluation of their suitability.

4.5.5.1.1 Recursive testing of subtype Relationships

It is possible to determine whether one *concept* subsumes another *concept* by recursively following every possible sequences of | is a | *Relationships* from a candidate *concept* until the predicate *concept* is reached or until all possible paths have been exhausted.

This approach is far too slow to deliver effective implementations in all environments in which it has been tested to date.

4.5.5.1.2 Semantic type Identifiers and hierarchy flags

Flags added to the internal representation of each *Concept* can be used to indicate the set of high-level *concept* nodes of which that *concept* is a *subtype*. A *concept* can only subsume *concepts* that include the same set of high-level *concept* flags. This approach can reduce the number of tests that need to be performed to recursively test the *subtype relationships*:

- If a candidate does not have all the high-level node flags that the predicate has, no further tests are needed. The candidate is not a *subtype* of the predicate.
- Even if a candidate shares the high-level node flags with the predicate, any path that reaches a *concept* that does not share those flags need not be further tested.

While faster than the unaided recursive testing approach, this is too slow to deliver effective implementations and is not scalable.

4.5.5.1.3 Use of proprietary database features

Some databases include additional features to support the recursive testing of a chain of hierarchical *relationships*. Other methods of optimization that may be applied to allow more rapid computation of *subtype descendant relationships* are outlined in the following subsections.

Current experiences of databases that support this type of approach indicate that (while easy to implement) the performance is substantially inferior to use of branch-numbering or *transitive closure* (see *Precomputed Transitive Closure table*).

4.5.5.1.4 Branch numbering

The internal representation of each *Concept* can be extended to include a branch-number and a set of branch-number -ranges.

A branch-numbering algorithm can then be applied when each release of *SNOMED CT* is imported.

A typical branch-numbering algorithm processes the *subtype hierarchy* in the following way:

- A depth first tree walk is performed starting from the root *Concept* (branch-number 1) and an incrementing number is applied to each *Concept* when it is encountered for the first time.
- After the branch numbers have been computed a further tree walk allocates one or more branch-number ranges to each *Concept* with any *subtype descendants*:
  - Many *Concepts* will have a single branch number range containing all their *descendants*.
  - Some *Concepts* will have several non-contiguous ranges of *descendant Concept* branch numbers:
    - This is because a *Concept* may have multiple supertypes. Therefore, the *descendants* of a *Concept* may have branch numbers that were allocated as a result of their *relationship* to another *ancestor Concept*. However, the path from any *Concept* to the root *Concept* always converges at or before the top-level *Concept*. Therefore, multiple ranges coalesce when reaching more general common *supertype ancestors*.
- At run time, rather than needing to traverse many *subtype Relationships*, the branch number of each *Concept* is tested for inclusion in the branch number range of the putative *ancestor*.

This approach removes the need for exhaustive testing of *subtype Relationships*. The disadvantages are a relatively complex build process that must be repeated for each release or update and a requirement for the internal *Concept* representation to accommodate a variable length representation of branch number ranges.

4.5.5.1.5 Precomputed Transitive Closure table

The *transitive closure* table is a comprehensive view of all the supertypes of every *concept*. It can be derived from *current* release data by traversing all | is a | *relationships* recursively and adding each inferred supertype *relationship* to a table.

The advantage of this type of view is that a *candidate - concept* can be tested for subsumption by *predicate - concept* by a simple SQL *query*. In addition, the table can be updated to take account of changes without requiring a complete rebuild. The disadvantage is the storage capacity required.

Note: The *transitive closure* table for the active content of the current version of the *International Release*, has about six million rows. The row count increases when *Extensions* are included. Typical database representations of the *transitive closure* table and associated indexes consume more than a Gigabyte of disk storage.

4.5.5.1.6 Recommendations

The *Transitive Closure* method is strongly recommended for use in any environment requiring high performance where disk capacity for storage and/or bandwidth for distribution are not a problem.

Where disk capacity and/or distribution bandwidth are limiting factors, Branch Numbering provides an efficient alternative approach.

4.5.5.2 Transitive closure implementation

4.5.5.2.1 Transitive closure distribution

It has been proposed that in future a *transitive closure* table would be released.
This would support easier implementation and provide a reference against which to check alternative algorithms. The format for such a distributed *transitive closure* table is the subject of separate documents that are under development.

The following sections provide basic advice on generating and using a simple functional *transitive closure* table. Even if the *SNOMED CT International Release transitive closure* is distributed, there may be a requirement for implementers to generate a *transitive closure* that includes additional content from one or more *SNOMED CT Extensions*.

4.5.5.2.2 Transitive closure table structure

The simplest form for a *transitive closure* table has two columns labeled "SubtypeId" and "SupertypeId". Each of these columns has a datatype that supports the *SNOMED CT Identifier* and is populated by *concept identifiers*.

This simple table requires one unique index "SubtypeId+SupertypeId" and a secondary non-unique index by "SupertypeId" to allow efficient reversed lookup.

Additional columns may be included to optimize some extended functionality. For example:

- A flag to indicate rows that represent links between a *concept* and its proximal *primitive* supertypes.
- If *inactive concepts* are included in the table, a flag to indicate the nature of any *historical relationship* traversed.
- A semantic distance count indicating the number of direct **|** is a **|** *relationship* between the *subtype* and supertype. Although such a number has not absolute meaning it may be useful as a relative measure of proximity.
- An *Identifier* of the *transitive closure* row. This may be of value for maintaining history of changes to *transitive closures* between releases.

4.5.5.2.3 Generating a transitive closure table

There are various ways in which a *transitive closure* table can be generated. The method illustrated here represents the smallest SQL *query* that might be used for this purpose. It is not the most efficient *query* and may take several hours to run. However, it does offer a simple standard for comparing the results of alternative approaches.

Table 259: Sample SQL for Deriving a Transitive Closure Table

-- Create simple sct_transitive closure table -- This version uses varchar for *string ConceptId* values -- An alternative would be to use bigint using the 64-bit *integer* representation CREATE TABLE [sct_transitiveclosure] ([SubtypeId] [varchar] (18) COLLATE Latin1_General_CI_AS NOT NULL , [SupertypeId] [varchar] (18) COLLATE Latin1_General_CI_AS NOT NULL , CONSTRAINT [PK_sct_transitiveclosure] PRIMARY KEY CLUSTERED ([SubtypeId] [SupertypeId],))
-- Add inverted index CREATE INDEX [IX_sct_transitiveclosure] ON [dbo].[sct_transitiveclosure]([SupertypeId])
-- Initial row creation self reference INSERT INTO [sct_transitiveclosure]([SupertypeId], [SubtypeId]) SELECT [*ConceptId*], [*ConceptId*] FROM [sct_concepts]
-- Further row creation WHILE @@ROWCOUNT>0 -- Repeats while new rows are created by the following *query* INSERT INTO [sct_transitiveclosure] ([SupertypeId], [SubtypeId]) SELECT distinct [*ConceptId2*],[tc].[SubtypeId] FROM [sct_relationships] INNER JOIN [sct_transitiveclosure] as tc ON [*ConceptId1*]=[tc].[SupertypeId] LEFT OUTER JOIN [sct_transitiveclosure] As tc2 ON [*ConceptId2*]=[tc2].[SupertypeId] AND [tc].[SubtypeId]=[tc2].[SubtypeId] WHERE [RelationshipType]='116680003' AND [tc2].[SubtypeId] is null

4.5.5.2.4 Using the transitive closure table to check subsumption

The following SQL *query* illustrates a simple way to use the *transitive closure* table for testing subsumption. In practice , this type of clause would be included in a more complex *query* allowing many candidates and predicates to be tested as a condition of retrieval in a single *query*.

Table 260: Sample SQL for Applying a Transitive Closure Table

SELECT * FROM TransitiveClosure AS tc WHERE [tc].[Supertype]= [predicate - *concept*].[conceptId] AND [tc].[SubtypeId]=[candidate - *concept*].[conceptId]

Unless storage capacity is a significant *constraint*, a pre-computed *transitive closure* table appears to out-perform other options and is robust, flexible and easy to implement.

4.6 Supporting Selective Data Retrieval

This section addresses the types of *terminology service* that are required to enable effective use of the *SNOMED CT* hierarchies and definitions when retrieving data.

Please refer to Supporting Selective Data Retrieval in the revised Terminology services Guide for Release Format 2.

4.7 Terminology Server Software

This section outlines the possible characteristics of software that provides *Terminology services* through a programmable interface. Such software represents an approach to development that may enable more rapid implementation of *SNOMED CT*.

This guide does not specify a particular *Application Programming Interface* (*API*) for accessing *SNOMED CT* services. Instead it sets out the general principles and options for delivery and use of a *terminology server*.

4.7.1 Terminology server functionality

A *terminology server* should be able to deliver all the essential *Terminology services* identified in the *Terminology Services Guide (RF1) (4)*. It should also provide the recommended *Terminology services* and should achieve a performance that meets the more general requirements for the functionality of *SNOMED CT enabled applications*.

Terminology server may provide two types of service:

- Reference Services (see *Figure 53*):
  - Services that do not include a *user interface*;
  - The client application may use reference services to undertake many different functions;
  - For some of these functions the client application will populate an appropriate *user interface* component.

Example:

A reference server may return a list of *Descriptions* matching a particular search *string*. The client application may use this data to populate a list from which a user makes a selection.

- *User Interface* (*UI*) Services (see *Figure 54*):
  - Services that include the one or more *user interface* components that can be used in and programmatically accessed by the client application.

Example:

A *UI* server may provide a control that includes a text box and a list. When the user types in the text box, the server populates the list and allows the user to select an item. The selected item is accessible from the client program.

- One possible type of *UI* service is a *SNOMED CT browser* with an *API* for returning selected data to a client application:
  - This may be useful as mechanism for providing some *SNOMED CT* capabilities to an application. However, it is less suitable for frequent entry of *SNOMED CT* encoded information.

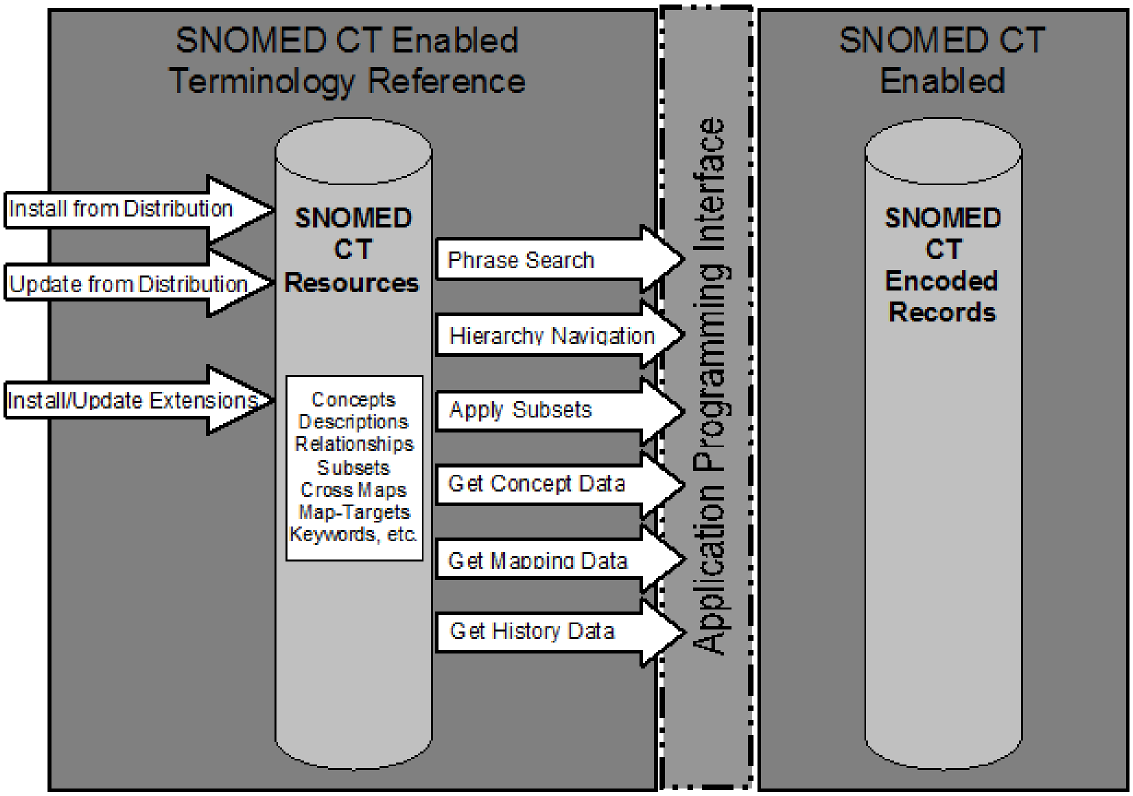


Figure 53: Terminology server providing reference services to a client application

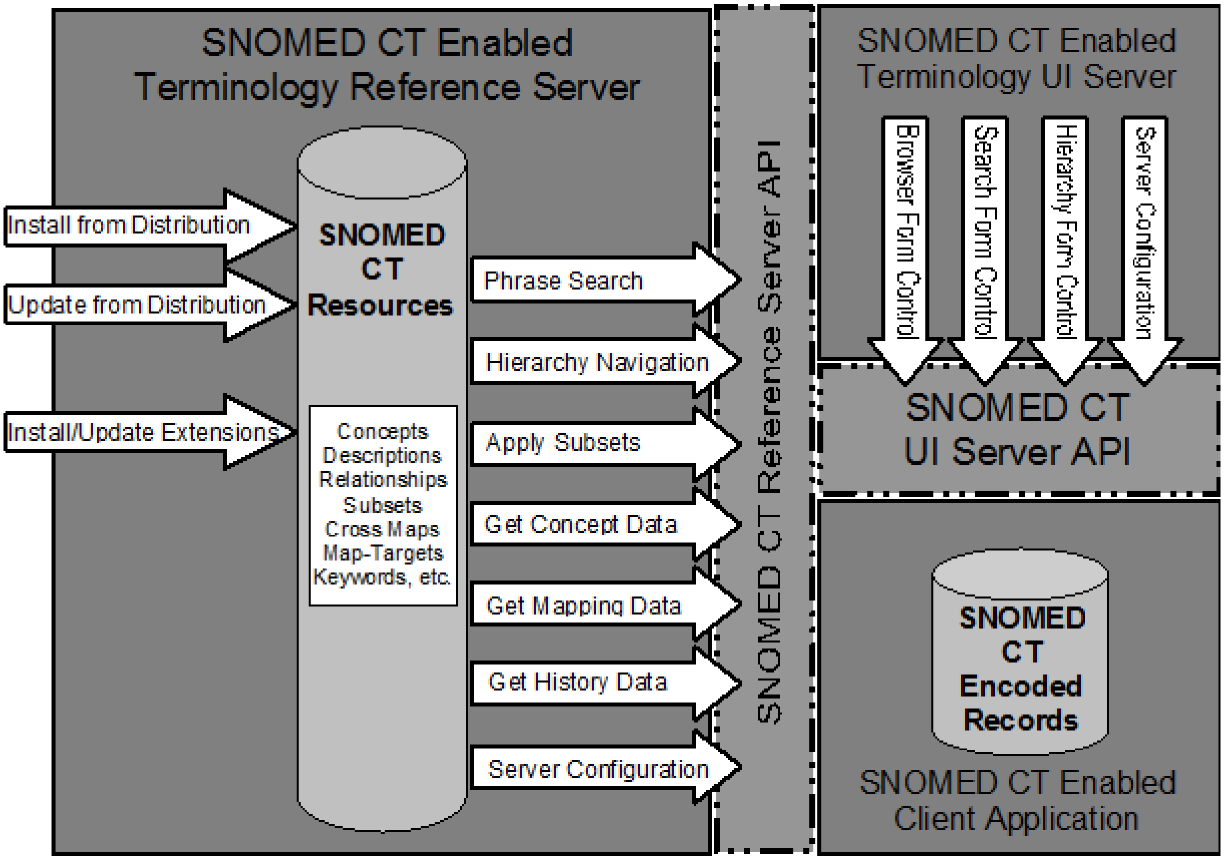


Figure 54: Terminology server providing user-interface and reference services to a client application

4.7.2 Terminology server APIs

This guide does not specify a particular *API*. The services specified in this guide may be delivered using various types of interfaces based on a range of different technologies including:

- Web services such using WSDL (Web Services *Description Language*) or REST (Representational State Transfer) interfaces;
- Java components such as JavaBeans™ or Eclipse plug-ins;
- Microsoft .NET® or Active -X® / COM / DCOM in Microsoft Windows® environments;
- CORBA® (Common Object Request Broker Architecture).

Decisions on which technologies to support depend on the intended functionality, performance, accessibility, ease of use and support requirements for maintenance or updates.

Over the past two decades there have been various efforts to specify standards for *terminology servers* and related *APIs*. The most recent development in this area is centered around the Common *Terminology Server Release* 2 (*CTS2*). The requirements initially identified and documented within *HL7* have now led to an OMG (Object Management Group) proposal. At least one of the responses to this proposal focuses directly on *SNOMED CT* related requirements. The OMG process is expected to result in a detailed specification and prototype implementation during 2011.

Chapter 5

5 SNOMED CT Release Format 1 file and field names

This section lists the file and field names used in *Release Format 1* 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..

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

Canonical table (RF1)

A table that contains the *Canonical form expressions* for all *SNOMED CT Concepts*. This table contains some (but not all) of the attributes present in the *Relationships Table*. It only contains the *Defining characteristics* required to distinguish a *Concept* from its most proximate *Primitive* supertype *Concepts*. The set of | is a | *subtype Relationships* excludes *Relationships* to *Fully defined* (non - *primitive*) *Concepts* and includes instead appropriate additional *Relationships* to the most proximate supertypes.

ChangeType (field in RF1)

A field in the *Component History Table* that indicates the nature of a change to a *component* made in a specified release of *SNOMED CT*.

CharacteristicType (field in RF1)

A field in the *Relationships Table* that indicates whether a *Relationship* specifies a *defining characteristic*, a *qualifying characteristic*, a *historical relationship*, or an additional non - *defining characteristic*.

Note: Field name in SNOMED CT Release Format 1

Component history table (RF1)

A data table consisting of rows each of which represents an item of *Component History*. The *Component History Table* is part of the *History Mechanism*.

ComponentId (field in RF1)

A general *term* used to refer to the primary *Identifier* of any *SNOMED CT Component*. *ComponentIds* include *ConceptIds*, *DescriptionIds*, *RelationshipIds*, *SubsetIds*, CrossMapSetIds and *TargetIds*. All ComponentIdsfollow the form of the *SCTID* specification.

ConceptId (field in RF1)

A *SNOMED CT Identifier* that uniquely identifies a *Concept* (meaning).

Example: For the meaning named | Pneumonia (disorder) |, the *ConceptId* is 233604007.

Note: Field name in SNOMED CT Release Format 1

ConceptId1 (field in RF1)

A field in the *Relationships Table* and *Canonical Table* that refers to the first of two related *Concepts*. The first *Concept* is defined or qualified by a *Relationship* to the second *Concept*.

ConceptId2 (field in RF1)

A field in the *Relationships Table* and *Canonical Table* that refers to the second of two related *Concepts*. The second *Concept* defines or qualifies the first *Concept*.

Concepts table (RF1)

A table that includes all *SNOMED CT concept* codes. Each *concept* code is represented by a single row.

ConceptStatus (field in RF1)

A field in the *Concepts Table* that specifies whether a *Concept* is in *current* use. Values include "*current*", "duplicate", "erroneous" "ambiguous" and "limited".

Context concept subset (RF1)

A *Subset* used to specify the *Concepts* that form part of a context - domain.

Context description subset (RF1)

A *Subset* used to specify the *Descriptions* that form part of a context - domain.

ContextId (field in RF1)

A field in the *Subsets Table* that specifies the context within which a *Context Concept Subset* or *Context Description Subset* is valid.

Core table (RF1)

Refers to the *SNOMED CT Concept*, *Relationship* and *Description* Tables.

Alternatives

SNOMED CT core table

Cross Map (RF1)

A *Cross Map* is a reference from a *Concept* code to a *Cross Map Target*. Each *Cross Map* is represented as a row in the *Cross Maps Table*. It links a single *SNOMED CT concept* code to one or more codes in a target classification (such as ICD-9-CM) or terminology. A *Concept* code may have a single *Cross Map* or a set of alternative *Cross Maps*

Cross Map set (RF1)

A set of *Cross Maps* that together provide a valid way of mapping some or all *SNOMED CT Concepts* to a specified *Target Scheme*. Alternative *Cross Map Sets* may exist for the same *Target Scheme*, if business rules or guidelines alter the appropriateness of particular mappings to that scheme. Each *Cross Map Set* is represented as a row in the *Cross Map Sets Table*.

Cross Map sets table (RF1)

A data table consisting of rows each of which represents a *Cross Map Set*.

Cross Map target (RF1)

A code or set of codes in a *Target Scheme* that together represent an appropriate mapping from a clinical statement expressed using *SNOMED CT*. Some *Cross Map Targets* may be derived from two or more associated statements and in these cases the combination can be expressed as a set of associated rules. Each *Cross Map Target* is represented as a row in the *Cross Map Targets Table*.

Cross Map targets table (RF1)

A data table consisting of rows each of which represents a *Cross Map Set*.

Cross Maps table (RF1)

A data table consisting of rows each of which represents a *Cross Map*.

CTV3ID (field in RF1)

A five-character code allocated to a meaning or term in *Clinical Terms Version 3* (*CTV3*, previously known as *Read Codes*). Each row in the *SNOMED CT concepts* table has a field for the corresponding *concept* code from *CTV3*.

Note: Field name in SNOMED CT Release Format 1

Note: The *CTV3ID* and *SNOMEDID* fields are no longer supported in *Release Format 2*. Instead a |*Simple map (reference set)*| is used to document the link between legacy codes and *SNOMED CT*.

Note: The *CTV3ID* field should no longer be relied upon for mapping to and from the *Read Codes*. Additional mapping work in the UK identified some anomalies and resulted development of more flexibility table for *Read Code* Mapping

Current

A *ConceptStatus* that implies a *component* appropriate for use and is not subject to a planned move to another *Extension Namespace*.

D

DescriptionId (field in RF1)

A *SNOMED CT Identifier* that uniquely identifies a *Description*.

Descriptions table (RF1)

A data table consisting of rows, each of which represents a *Description*.

DescriptionStatus (field in RF1)

A field in the *Descriptions Table* that specifies whether a *Description* is in *current* use. Values include "*current*", "duplicate", "erroneous" "inappropriate" and "limited".

DescriptionType (field in RF1)

A field in the *Descriptions Table* that specifies whether a *Description* is a *Fully Specified Name*, *Preferred Term*, or *Synonym*. The *DescriptionType* is *language* dependent and may be changed by applying a *Language Subset*. It may be "undefined" in the released *Descriptions Table* in which case the *Description* is not used unless an appropriate *Language Subset* is applied.

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 *Word Search Tables - Summary*].

Note: File or Table name in SNOMED CT toolkit

Duplicate terms subset (RF1)

A *Subset Type* that identifies *Duplicate Terms* and allows a priority to be specified between these for use in searches.

E

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 *Word Search Tables - Summary*].

Note: File or Table name in SNOMED CT toolkit

F

FullySpecifiedName (field in RF1)

A *term* unique among *active Descriptions* in *SNOMED CT* that names the meaning of a *Concept* code in a manner that is intended to be unambiguous and stable across multiple contexts.

H

Historical relationship (RF1)

A *Relationship* that refers from an *Inactive Concept* to an *Active Concept* that duplicates, corrects, replaces or disambiguates it. Note that *Historical Relationships* are used in a way similar to References for *Descriptions*. However, as part of the *Relationships Table* they are more readily accessible for computation and retrieval of legacy data.

History mechanism (RF1)

The *history mechanism* is the information distributed with *SNOMED CT* designed to track the history of changes to its logic definitions and *descriptions*. The *history mechanism* is supported by two distribution tables:

- *Component History Table*
- *References Table*

I

InitialCapitalStatus (field in RF1)

A field in the *Descriptions Table* that specifies whether the capitalization of the first character is significant. If the value of this field is "1" then the first character should remain either in upper or lower case as released. Otherwise the case of the first character may be changed to suit its context in a sentence.

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.

IsPrimitive (field in RF1)

A field in the *Concepts Table* that indicates whether a *Concept* is *Primitive* or *Fully defined*.

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

Language subset (RF1)

SNOMED CT can be translated into virtually any human *language* or *dialect*. These translations attach new *language* -specific *terms* as *descriptions* of existing *concept* codes and may also use existing *descriptions* if translation is not necessary. A *language subset* is a set of references to the *descriptions* that are members of a *language* edition of *SNOMED CT*. Additionally, data in the *language subset* specifies the *DescriptionType* of each *description* (*Fully Specified Name*, *Preferred Term* or *Synonym*).

LanguageCode (field in RF1)

A field that indicates the *Language* and, optionally, *Dialect* applicable to a row in the *Subsets Table*, *Descriptions Table* or to an *Excluded Words Table*.

LinkedId (field in RF1)

A field in the *Subsets Table*.

M

MapAdvice (field in RF1)

A field in the *Cross Maps Table*, may contain human-readable advice on mapping.

MapConceptId (field in RF1)

A field in the *Cross Maps Table* containing the *Identifier* of the *Concept* that is the subject of the map.

MapOption (field in RF1)

A field in the *Cross Maps Table*, which specifies the *order* in which *Cross Maps* are tested for automated processing of *Cross Mapping* rules.

MapPriority (field in RF1)

A field in the *Cross Maps Table* that specifies which *Cross Maps* are most likely to apply to the associated *Concept*. The value 0 indicates a default map.

MapRule (field in RF1)

A field in the *Cross Maps Table* that may contain a computer processable representation of a rule that determines when this map should be used.

MapSetId (field in RF1)

A *SNOMED CT Identifier* that uniquely identifies a *Cross Map Set*.

MapSetName (field in RF1)

A field in the *Cross Map Sets Table* that names that *Cross Map Set*.

MapSetRealmId (field in RF1)

A field in the *Cross Map Sets Table* that indicates the *Realm* in which a set of *Cross Maps* is applicable.

MapSetRuleType (field in RF1)

A field in the *Cross Map Sets Table* that indicates whether any computer processable rules are present in the associated *Cross Maps* or *Cross Map Targets* and, if so, what form of *expression* is used to represent these rules.

MapSetSchemeId (field in RF1)

A field in the *Cross Map Sets Table* which identifies the classification or coding-scheme that is the target of a *Cross Map Set*.

MapSetSchemeName (field in RF1)

A field in the *Cross Map Sets Table* that contains the plain text name of the classification or coding-scheme that is the target of a *Cross Map Set*.

MapSetSchemeVersion (field in RF1)

A field in the *Cross Map Sets Table* that identifies the version of the classification or coding-scheme that is the target of a *Cross Map Set*.

MapSetSeparator (field in RF1)

A field in the *Cross Map Sets Table* that contains a character that acts as a separator between *target codes* in the *Cross Map Targets Table*.

MapSetType (field in RF1)

A field in the *Cross Map Sets Table* that indicates whether the *Cross Maps* associated with this *Cross Map Set* are all simple one to one maps or include, one to many and/or choices of alternative maps.

MapTargetId (field in RF1)

A field in the *Cross Maps Table*, which refers to the *TargetId* of a row in the *Cross Map Targets Table* that contains a *target scheme* mapping for a specified *Concept*.

MemberId (field in RF1)

A field in the *Subset Members Table*, which refers to the *ComponentId* of the *Concept*, *Description* or other *component* that is a member of a specified *Subset*.

MemberStatus (field in RF1)

A field in the *Subset Members Table*, which indicates the inclusion, exclusion, priority or *order* of an identified *Subset Member* in a specified *Subset*.

N

Navigation link (RF1)

An association between two *Concepts* that supports *Navigation* between *Concepts*. *Navigation Links* generate a *hierarchy* which has three distinct differences from the *subtype hierarchy* defined by |is a| *Relationships* this *hierarchy*: Does not contribute to the semantic definitions of *Concepts*; Specifies a display *order Concepts* within a set of *Concepts* linked to a common parent. Navigation Links are distributed as a *Navigation Subset*. Alternative *Navigation Subsets* may be specified and applied to vary the *navigation hierarchy* to meet the needs of particular groups of users.

Navigation subset (RF1)

A *Subset* that specifies sets of *Navigation Links* between *Concepts*.

R

Realm concept subset (RF1)

A *Subset* of *Concepts* applicable to a particular *Realm*.

Realm description subset (RF1)

A *Subset* of *Descriptions* applicable to a particular *Realm*.

Realm relationship subset (RF1)

A *Subset* of *Relationships* applicable to a particular *Realm* (for future use).

RealmId (field in RF1)

A field in the *Subsets Table* that identifies the *Realm* within which the specified *Subset* is applicable.

Reason (field in RF1)

A field in the *Component History Table* which provides a text *description* of the *reason* for a change in the *Status* of a *component*.

ReferencedId (field in RF1)

A field in the *References Table* that identifies the *current component* which replaces it or is duplicated by a non-*current component*.

References table (RF1)

A data table consisting of rows each of which represents a Reference between an *inactive component* and a *component* that was *active* for the release in which the first was made *inactive*. The *References Table* is part of the *History Mechanism*.

ReferenceType (field in RF1)

A field in the *References Table* that indicates whether a specified non-*current component* was replaced by, duplicated by, similar to or an alternative form of the referenced *current component*.

Refinability (field in RF1)

A field in the *Relationships Table*, which indicates whether it is permissible to refine the value of a *Defining characteristic* or *qualifying characteristic* to represent a more refined *Concept*.

Relationship group (RF1)

Field in the *Relationships Table* is used to group these *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.

RelationshipType (field in RF2)

The nature of a *Relationship* between two *Concepts*. *Relationship Types* are represented in *SNOMED CT* by *Concept* codes. In the *Relationships Table*, the RelationshipType field contains the *ConceptId* for the *concept* in *SNOMED CT* that forms the *relationship* between two other *concepts* (*ConceptId1* and *ConceptId2*). For defining and qualifying *relationships*, the *Relationship Type* is an *Attribute* code. RelationshipType should not be confused with *CharacteristicType*.

RelationshipId (field in RF2)

A *SNOMED CT Identifier* that uniquely identifies a *Relationship*. *RelationshipId* is the key of the *Relationships Table*. Each row in the *Relationships Table* represents a *relationship* triplet (*ConceptId1* RelationshipType - *ConceptId2*).

Relationships table (RF1)

A data table consisting of rows, each of which represents a *Relationship*.

Retired concept (RF1)

A *Concept* that has been made *inactive* with no *reason* specified. *Concepts* that are no longer *current* should be called "non-*current*" or "*inactive*" rather than " *retired*." (See *Inactive Concept*.)

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.

SNOMEDID (field in RF1)

A field in the *Concepts Table* that contains the legacy SNOMED International 3 to 3.5 code for the *Concept*.

Status

The *Status* of a *component* indicates whether it is in *current* use and, if not, provides a general indication of the *reason* that it is not recommended for *current* use. The *Status* of a *Concept* is referred to as *ConceptStatus* and the *Status* of a *Description* is referred to as *DescriptionStatus*.

Note: RF1 specific - replaced in RF2 by "active" attribute for all components.

String (data type)

A datatype representing a sequence of characters.

Note: In *SNOMED CT release file* specifications strings are represented using *Unicode UTF-8* encoding.

Subset definition (RF1)

A series of clauses that specifies the nature of a *Subset* and determines its membership. A *Subset Definition* is an alternative form of representation applicable to many *Subsets*. See also *Subset Definition File*.

Subset definition clause (RF1)

A statement that refers directly or indirectly to one or more *Concepts*, *Descriptions* or *Relationships* and indicates whether the referenced item(s) should be removed from or added to a specified *Subset*. In the case of additions to a *Subset* the clause also specifies the *MemberStatus* to be assigned to those additions.

Subset definition file (RF1)

A file containing a series of clauses that define the nature and membership of a *Subset*. The *Subset Definition File* is an XML document that contains the *Subset Definition* for a single *Subset*. A *Subset Definition File* can be used to generate the appropriate *Subsets Table* row and *Subset Members Table* rows to represent a *Subset*. Therefore applications need not support this alternative form of *Subset* representation.

Subset member (RF1)

A *Concept*, *Description*, *Relationship* or another *Subset* that is part of a specified *Subset*.

Subset members table (RF1)

A data table consisting of rows each of which refers to a single *Subset Member*.

SubsetVersion (field in RF1)

A version number assigned to a particular release of a *Subset*.

SubsetId (field in RF1)

A *SNOMED CT Identifier* that uniquely identifies a *Subset*.

SubsetName (field in RF1)

A field in the *Subsets Table* that contains a human readable name for the *Subset*.

SubsetOriginalId (field in RF1)

A field in the *Subsets Table* that identifies the first version of the *Subset* on which this *Subset* is based. For the first version of a *Subset* the *SubsetOriginalId* and *SubsetId* fields contain the same value. For each subsequent version the *Subset Version* is incremented and a new *SubsetId* is allocated but the *SubsetOriginalId* field retains the same value in all versions.

Subsets table (RF1)

A data table consisting of rows each of which represents a *Subset*.

SubsetType (field in RF1)

An indication of the type of component that may be a member of a *Subset*.

Alternatives

Subset type
Type of subset

T

TargetAdvice (field in RF1)

A field in the *Cross Map Targets Table* that may contain human readable advice on the circumstances in which this *Cross Map Target* is applicable.

TargetCodes (field in RF1)

A field in the *Cross Map Targets Table* that contains one or more codes or *Identifiers* in the *target scheme* or classification. If there is more than one *Target Code* they are separated by a separator specified in the associated row of the *Cross Map Sets Table*.

TargetId (field in RF1)

A *SNOMED CT Identifier* that uniquely identifies a *Cross Map Target*.

TargetRule (field in RF1)

A field in the *Cross Map Targets Table* that may contain computer processable rules specifying the circumstances in which this *Cross Map Target* is applicable.

TargetSchemeId (field in RF1)

A field in the *Cross Map Targets Table* that identifies the target coding scheme or classification to which this *Cross Map Target* applies.

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**

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.

UTF-8

A standard method of encoding *Unicode* characters in a way optimized for the ASCII character set. *UTF-8* is described in [see *Unicode UTF-8 encoding* on page 89].

Note: This encoding is used for release file fields of data type "String".

UUID (data type)

A datatype representing a sequence of unique *Identifier* encoded as a 128-bit integer.

Note: In *SNOMED CT release files UUIDs* are represented using 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

Alternatives

Universally Unique Identifier
GUID
Globally Unique Identifier

W

Word equivalents table

A data table in which each row represents a *Word Equivalent*. See [see *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 *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

Chapter 6

6 Glossary

This section contains selected terms from the IHTSDO Glossary. The full IHTSDO Glossary is available as follows:

- Online access: *www.ihtsdo.org/gl*;
- PDF file (US English): *www.ihtsdo.org/gl.pdf*;
- PDF file (GB English): *www.ihtsdo.org/gl_gb.pdf*.

A

Active component

A *SNOMED CT component* that is intended for use. *Release files* contain *Active* and *Inactive components* to provide a historical record of the content of the terminology at different points in time.

Note: A component is active when the most recent row with the relevant Component.*id* in the *Full Release* of the relevant *Release File* has the value Component.*active*=1 (one). The most recent row for a component is determined based on the Component.*effectiveTime* value.

Active concept

A *Concept* that is intended for use. *Release files* contain *Active* and *Inactive components* to provide a historical record of the content of the terminology at different points in time.

Note: A component is active when the most recent row with the relevant Component.*id* in the *Full Release* of the relevant *Release File* has the value Component.*active*=1 (one). The most recent row for a component is determined based on the Component.*effectiveTime* value.

Active description

A *Description* that is intended for use. *Release files* contain *Active* and *Inactive components* to provide a historical record of the content of the terminology at different points in time.

Note: A component is active when the most recent row with the relevant Component.*id* in the *Full Release* of the relevant *Release File* has the value Component.*active*=1 (one). The most recent row for a component is determined based on the Component.*effectiveTime* value.

Affiliate

An *IHTSDO Affiliate Licensee* in accordance with the *IHTSDO* Affiliate License Agreement.

Alternatives

IHTSDO Affiliate
Affiliate Licensee

Affiliate Licence Agreement

The agreement between an *IHTSDO affiliate* (the licensee) and the *IHTSDO* (the licensor) under which developers and implementers are permitted to use the *SNOMED CT International Release* and distribute it to their sub-licensees as part of a software system.

Alternatives

Affiliate Licence

ANSI

American National Standards Institute (*ANSI*) is a private non-profit organization that oversees the development of voluntary consensus standards for products, services, processes, systems, and personnel in the United States. The organization also coordinates U.S. standards with international standards.

Alternatives

ANSI
American National Standards Institute

Application Programming Interface

Application Programming Interface

A set of rules and specifications that enable communication between software programs. *Application Programming Interfaces* enables interaction between separate software programs, in much the same way that a *user interface* facilitates interaction between humans and computers.

Alternatives

API

Attribute

An *attribute* represents a characteristic of the meaning of a *concept* or the nature of a refinement.

Note: An *attribute* has a name which is represented by a *concept*. All the *concepts* that can be used to name *attributes* are *subtypes* of the *concept* | *concept model attribute* |. An *attribute* is assigned a value (*attribute value pair*) when used in the definition of a *concept* or in a *postcoordinated expression*. The permitted *attribute values* (*range*.) for an *attribute* depend on the *attribute name* and on the *domain* of the *concept* being refined.

Example: 116676008 | Associated morphology |

Alternatives

Concept Model Attribute
Relationship Type
Role

Attribute group

An association between a set of *attribute value* pairs which causes them to be treated separately from other *attribute value* pairs in the same definition or *postcoordinated expression refinement*.

Example:

The definition of the *concept* |cholecystectomy with exploration of common duct| has two |method| attributes with different values (|excision -action| and |exploration -action|) and two |procedure site direct| attributes with different values (|common bile duct structure| and |gallbladder structure|). The attributes are grouped so that procedure is not incorrectly classified as an |excision of common bile duct|.

Alternatives

AttributeGroup

Attribute name

A *concept* that represents the type of a *relationship* or the type of a *refinement* in a *postcoordinated expression*.

Notes:

1. The type of a *relationship* is indicated by the *typeId* attribute in the *Relationship file*
2. The *concepts* that can be used to name attributes are:
   - 116680003 | Is a (attribute) | and
   - *subtypes* of 410662002 | Concept model attribute |

Alternatives

Relationship Type
AttributeName

Attribute value

A *concept* that represents the target of a *relationship* or the value of an *expression refinement* in a *postcoordinated expression*.

Alternatives

Attribute-value
AttributeValue

Attribute value pair

A combination of an *attribute name* and an *attribute value* used to represent a specific type of information in a generic way without altering the underlying structure of an information model. The *attribute name* identifies the type of information and the *attribute value* provides a value.

Note: *Attribute value pairs* are used by *SNOMED CT* in *relationships* and *postcoordinated expressions*. In both cases, the *attribute name* and *attribute value* are expressed using *SNOMED CT concept identifiers*. In the *Relationship file*, the *attribute name* if represented by the *Relationship.typeId* and the *attribute value* by the *Relationship*. *destinationId*.

Authoritative concept

A *concept* with a specific meaning defined by an authoritative source such as a national or international professional body or standards organization.

Authorized Triage Organization

An organization approved by the *IHTSDO* to manage and triage change requests to for inclusion of content in the *SNOMED CT International Release* and/or one or more National *Extensions*.

Note: *IHTSDO Members* and their *National Release Centers* are likely to fulfill this role. In addition, *IHTSDO affiliates* and Standards Development Organizations may be eligible for consideration as *Authorized Triage Organizations*.

Alternatives

ATO

Automatic classification

A process that generated a logically consistent *subtype classification* by applying *description logic* rules to the stated definitions of a set of *concepts*.

Alternatives

Auto classify

B

Baseline

A release status applied to a collection of *SNOMED CT release files* that represent the first formally endorsed release of additions of *components* and/or *derivatives* to the *SNOMED CT International Release* or to a *SNOMED CT Extension*. The *Baseline* status indicates the releasing party (*IHTSDO* or the owner of the *Extension*) commits to maintain the release history of this release and all subsequent updates. Once confirmed as a *Baseline*, additional *components* and *derivatives* will be maintained and versioned in accordance with the Release Format 2 specification (i.e. by adding rows to the *Full Release* with the *effectiveTime* appropriate to the update).

Note: The significance the *Baseline* status is that implementers can use the additional *components* and *derivatives* in operational systems, with confidence in the subsequent maintenance of these additions.

Browser

A computer application or software tool used for exploring and searching terminology content. A typical *SNOMED CT browser* can locate *concepts* and *descriptions* by *Identifiers* and by searching the text of *description terms*. Various views of located *concepts* may be displayed including the set of related *descriptions*, the hierarchical *relationships* and other defining *relationships*.

Alternatives

SNOMED CT browser

C

Candidate Baseline

A provisional status applied to a collection of *SNOMED CT release files* that represent a proposed additions of *components* and/or *derivatives* to the *SNOMED CT International Release* or to a *SNOMED CT Extension*. The *Candidate Baseline* status indicates the releasing party (*IHTSDO* or the owner of the *Extension*) expects to subsequently confirm the release as the *Baseline*. However, if a significant issue is reported in its format or content, the releasing party reserves the right to withdraw a *Candidate Baseline* release, or to replace it with another *Candidate Baseline*, to address the issue. The releasing party need not commit to this being an actual *Baseline* release until shortly before the due date for the next release.

Note: The significance the *Candidate Baseline* status is that anyone implementing this data must be prepared for withdrawal or significant changes that may occur to the additional *components* or *derivatives*. Therefore, this data should not be used in an operational environment in ways that create a dependency on continued maintenance of the additional *components* or *derivatives*. However, a *Candidate Baseline* may be confirmed as the *Baseline* and, in that case all subsequent updates to the additional *components* and *derivatives* will be fully version tracked from the release of the *Candidate Baseline*.

Canonical form

An serialized representation of a *SNOMED CT expression* which follows the *normal form* and in which the *refinements*, *attributes* and *attribute groups* are arranged in a standard order.

Cardinality

? A measure of the number of elements in a set. Modeling rules include constraints on the *cardinality* of particular attributes or associations between classes.

CEN

The European Committee for Standardization is a major provider of European Standards and technical specifications. Its mission is to foster the European economy in global trading, the welfare of European citizens and the environment. Through its services it provides a platform for the development of European Standards and other technical specifications.

Alternatives

Comité Européen de Normalisation
European Committee for Standardization
Europäisches Komitee für Normung

CEN TC251

CEN /TC 251 (*CEN* Technical Committee 251) is a committee within the European Committee for Standardization (*CEN*) working on standardization in the field of Health Information and Communications Technology (ICT) in the *European Union*. Its goal is to achieve compatibility and interoperability between independent systems and to enable modularity in *Electronic Health Record* systems.

Check digit

The *check-digit* is the final (rightmost) digit of the *SNOMED CT Identifier* (*SCTID*). It can be used to check the validity of *SCTIDs*. *Clinical Information Systems* can use the *check-digit* to identify *SNOMED CT* codes that have been entered incorrectly (typo errors, etc). It is calculated using the Verhoeff algorithm.

Clinical Information System

A computer-based system that is designed for collecting, storing, manipulating and making available clinical information to support the delivery of healthcare services to individual people and populations.

Alternatives

CIS

Clinical Terms Version 3

One of the source terminologies, along with *SNOMED RT*, that were used to develop *SNOMED CT*. *CTV3* is UK Crown Copyright, distributed by the United Kingdom *National Health Service* (*NHS*), and is integrated into *SNOMED CT*.

Alternatives

CTV3
Version 3 of the Read Codes

C-NPU

Nomenclature, Properties and Units (*C-NPU* in collaboration with International *Union* of Pure and Applied Chemistry (IUPAC) The *IFCC-IUPAC* coding system Provides a terminology for Properties and Units in the Clinical Laboratory Sciences

Alternatives

Nomenclature, Properties and Units
NPU
IFCC IUPAC

Note: The name of the organization responsible for C-NPU sometimes used as a synonym

Collaborative Space

A web resource with software to help people involved in a common task achieve goals by enabling effective communication within an project or organization.

Note: The *IHTSDO Collaborative Space* supports the communication needs of *IHTSDO* governance and advisory bodies. *IHTSDO* Standing Committees, Affiliate Forum, Member Forum and Working Groups all have *Collaborative Space* Projects each of which contain meeting announcements, discussions, shared documents and issue trackers.

Alternatives

Collabnet

Common Terminology Services 2

An *Application Programming Interface* (*API*) specification that is intended to describe the basic functionality that needed by healthcare software implementations to query and access terminological content. *CTS2* defines the functional requirements of a set of service interfaces to allow the representation, access, and maintenance of terminology content either locally, or across a federation of *terminology service* nodes.

Note: *CTS2* is specified as an *API* rather than a set of data structures to enable a wide variety of terminological content to be integrated within a common framework without the need for significant migration or rewrite.

Note: *CTS2* was developed from the original the [see *HL7 CTS specification*] and is now a joint initiative between HL7 and the [see *Object Management Group (OMG)*].

Alternatives

CTS2
HL7 CTS2

Complement

In set theory the *complement* of set A relative to the universal set U is the set of all members of U that are not members of A.

Note: Set theory is applied when describing the intended result of combinations of Reference Sets or Constraints.

SNOMED CT Component

Refers to any item identified by an *SCTID* in the main body of *SNOMED CT*, or in an authorized *Extension*. The *partition-identifier* indicates the type of component referred to by that *SCTID*. Each *component* is a uniquely identifiable instance of one of the following:

- *Concept*
- *Description*
- *Relationship*

Alternatives

Component

Component history

A record of an addition or change in the *status* of a *SNOMED CT Component* in a particular *Release Version*.

Compositional grammar

The set of rules that govern the way in which *SNOMED CT expressions* are represented as a plain text string.

Note: The specification of the [see *SNOMED CT Compositional Grammar*] is available as part of the Technical Implementation Guide.

Alternatives

SNOMED CT compositional grammar

Concept

A clinical idea to which a unique *Concept Identifier* has been assigned.

The *term concept* may also be used informally with the following meanings:

- The *concept Identifier*, which is the key of the *Concept file* (in this case it is less ambiguous to use the *term* "conceptId" or "*concept* code");
- The real-world referent(s) of the *Concept Identifier*, that is, the class of entities in reality that the *Concept Identifier* represents (in this case it is less ambiguous to use the *term* "meaning" or "code meaning").

Alternatives

SNOMED CT concept

Concept enumeration

Use of *SNOMED CT concept Identifiers* to represent of a set of values for a property of a particular type of *SNOMED CT component*.

Note: The *SNOMED CT concepts* used to represent *concept enumerations* are usually *subtype children* (or *descendants*) of a relevant general *concept* in the *SNOMED CT* metadata hierarchy. Each possible value is represented by a single child *concept*, and the set of values can be used to enable selection from a pick-list of one or more *concepts*.

Example:

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

Figure 55: Concept enumeration for: Description.typeId

Concept equivalence

Equivalence is the state of two *SNOMED CT concept* codes or *postcoordinated expressions* having the same meaning. *Concept equivalence* can occur when a *postcoordinated expression* has the same meaning as a *precoordinated concept* code; or when two different *postcoordinated expressions* have the same meaning.

Concept Identifier

A *SNOMED CT Identifier* that uniquely identifies a *Concept* (meaning).

Example: For the meaning named | Pneumonia (disorder) |, the *Concept Identifier* is 233604007.

Concept model

A set of rules that determines the permitted sets of *Relationships* between particular types of *concept*. The *Concept Model* specifies the attributes that can be applied to particular *concepts* and the ranges of permitted values for each of these attributes. There are also additional rules on the *cardinality* and grouping of particular types of *Relationships*.

Note: The [see *Concept Model Guide*] (which is part of the Technical Implementation Guide) summarizes the current set of rules applied to modeling *SNOMED CT concepts*. More detailed information, aimed at those involved creating and modeling content, is available in the *SNOMED CT* Editorial Guide.

Constraint

A rule that specifies limits on the attributes, values and associations that may be applied to a particular component.

Examples:

1. A modeling constraint may limit the permissible defining *Relationships* applied to a particular type of *concept*.
2. An instance data constraint may limit the permissible refinements that may be applied to particular *concept*

Context domain

A context domain is a set of values that are, or may be, used in an identifiable logical setting in an application, protocol, *query* or communication specification. A context domain may be very broad (e.g. procedures or diagnoses) or very narrow (e.g. procedures performed by a specialty or possible values for a field in specific message).

Context specific characteristic

A *Relationship* to a target *Concept* that provides information about the source *Concept* that is true at a particular time or within a particular country or organization . Contrast with *Defining characteristic* and *Qualifying characteristic*. Referred to in *CTV3* as a 'Fact'.

Context wrapper

The part of a *SNOMED CT expression* that specifies the context that applies to the *focus concept* that it contains.

Example: "Family history of asthma" can be represented by an *expression* in which the *concept* "asthma" is nested within an *context wrapper* that indicates that this is "family history" - rather than a current condition affecting the patient. For further details see [see *Modeling semantic context*].

Core file

A distribution file used to represent the main *SNOMED CT components* (*concepts*, *descriptions* and *relationships*).

Note: In the past the term "core" has also been used to refer to the content of the *SNOMED CT International Release* but this usage is deprecated.

Alternatives

SNOMED CT core
Core table
SNOMED CT core table
SNOMED CT core file
Core table

D

Darwin Information Typing Architecture

The Darwin Information Typing Architecture (*DITA*) is an XML-based architecture for authoring, producing, and delivering information. Although its main applications have so far been in technical publications, *DITA* is also used for other types of documents such as policies and procedures.

Note: *DITA* is used for creation, publication and maintenance of many *IHTSDO* guidance documents.

Alternatives

DITA

Data Analysis System

A computer system that is used to analyze records or other data that is encoded using *SNOMED CT*, but not if that system is also a *Data Creation System*;

Note: *IHTSDO* charges fees for use of *Data Analysis Systems* and *Data Creation Systems* in Non-Member Territories.

Data Creation System

A computer system that is used to create records or other data that is encoded using *SNOMED CT*.

Note: *IHTSDO* charges fees for use of *Data Analysis Systems* and *Data Creation Systems* in Non-Member Territories.

Data migration

Steps taken to enable legacy data to be accessible as part of a system that uses *SNOMED CT*.

Note: The objective of *data migration* is to enable data recorded prior to introduction of *SNOMED CT* can be retrieved and reused within a *SNOMED CT enabled application*. Options for *data migration* include actual conversion of the data or provision of methods for accessing the data in its original form.

Defining relationship

A *relationship* to a target *concept* that is always necessarily true from any instance of the source *concept*.

Example: The *defining relationships* of the concept | gastrectomy | include |method|=|excision - action| and |procedure site - Direct|=|stomach structure|.

Alternatives

Defining characteristic

Delta release

A *Release Type* in which the *release files* contain only component versions created since the previous release. Each component version in a *delta release* represents either a new component or a change to an existing component.

SNOMED CT Derivative

A document, subset, set of maps, or other resource that consists of, includes, references or is derived from one or more *SNOMED CT components*. The standard computer processable representation for most types of *SNOMED CT derivatives* is a *Reference set*.

Alternatives

Derivative

Description

An association between a human-readable phrase (*term*) and a particular *SNOMED CT concept* code. Each *description* is represented by a separate row in the *Description file*.

Note: Each *description* has a unique *identifier* and connects *concept* with a *term* of a specified *description type*.

Alternatives

SNOMED CT description

Description Identifier

A *SNOMED CT Identifier* that uniquely identifies a *Description*.

Description logic

A representation of semantic knowledge that allows formal reasoning to be applied based on axioms that state *relationships* between *concepts*.

Note: *Description logic* definitions of *SNOMED CT concepts* are represented by *defining relationships*. The formal rules of *description logic* can be applied to *defining relationships* by software tools (*description logic classifiers*) to interpret the meaning of *concepts*. This enables confirmation of the logical integrity of the terminology, and can also be used to support meaning-based retrieval from *SNOMED CT enabled* record systems.

Alternatives

DL

Related Links

Wikipedia entry on Description logic

Description logic classifier

A software tool that applies the rules of a *description logic* to a set of data to make inferences about the *relationships* between sets of *concepts*.

Note: *SNOMED CT concepts* and *relationships* are processed by a *description logic classifier* to generate the *subtype hierarchy*. *SNOMED CT expressions* can also be processed by a classifier to make inferences that support selective retrieval.

Alternatives

Classifier

Description Type

An indication of the intended usage of the *term* of a *SNOMED CT description* when applied to the associated *concept*.

Notes:

1. The *description type* is represented by the value of the *description.typeId* attribute.
2. Permitted values include the following (other types may be defined in future):

Table 261: Description types

typeId (with term)	Further information
900000000000003001 | Fully specified name |	A *term* unique among *active descriptions* in *SNOMED CT* that names the meaning of a *concept* code in a manner that is intended to be unambiguous and stable across multiple contexts (see *fully specified name (FSN)*).
900000000000013009 | Synonym |	A *term* that is an acceptable way to express a the meaning of a *SNOMED CT concept* (see *synonym*).
900000000000550004 | Definition |	An additional textual *description* applied to some *SNOMED CT concepts* that provides additional information about the intended meaning or usage of the *concept* (see *textual definition*).

3. The The *preferred term* is the *synonym* marked as preferred for use in the *Language reference set*for a given *language* or *dialect* (it is not a distinct *description type*).

Dialect

A *language* modified by the vocabulary and grammatical conventions applied to the *language* of a particular geographical or cultural environment.

Directed Acyclic Graph

A set of nodes connected to one another by lines (edges) in which each connection has a specified direction such that no route that follows the direction of the connections enters a loop (cycle).

Example: The *SNOMED CT subtype hierarchy* is an example of a *Directed Acyclic Graph*. *SNOMED CT concepts* are nodes and *"is a" Relationships* are the directed lines that connect them. All *"is a" Relationships* lead from a more specific *concept* to a more general *concept*, so a cycle would be a logical error (e.g. if "rubella virus" is a type of "virus" and "virus" is a type of "microorganism", then "microorganism" cannot be a type of "rubella virus").

Alternatives

DAG

Domain

A set of *concepts* which the *Concept Model* permits to be defined or refined using a particular set of *attributes* and *ranges*.

Note: A *domain* to which an *attribute* can be applied is typically defined to include concepts in one or more branches of the subtype hierarchy.

Example: The *domain* of the *attribute* 116676008 | Associated morphology | is defined as subtype of 404684003 | Clinical finding | hierarchy. Similarly, the *range* for values of 116676008 | Associated morphology | is subtypes of 49755003 | Morphologically abnormal structure |.

Alternatives

Concept model domain

Draft Standard for Trial Use

A *Draft Standard for Trial Use* is a specification and process to allow implementers to test a standard. At the end of the trial period the standard may be balloted, revised or withdrawn.

Example: The joint project between HL7 International and the *IHTSDO*, *TermInfo*, is an example of an HL7 DSTU.

Alternatives

DSTU

Duplicate term

A *Term* that occurs in several *Active Descriptions*. *Duplicate Terms* are valid in *SNOMED CT* since the intention is to provide natural *terms* used by clinicians rather than to apply formalized phraseology. The formalized form is provided by the *Fully Specified Name* and these are not permitted to be duplicated.

Dynamic snapshot view

A "*snapshot view*" for a specified date that is generated by filtering a "*full view*".

E

Electronic health record

A systematic collection of health information about individual patients or populations that is stored in a digital form. An *Electronic health record* may contain a complete and detailed record of a patient's health or may consist of a summary of information of particular relevance to continuing delivery of care.

Alternatives

EHR

EN13606

Electronic Health Record Communication (EN 13606) European Standard developed by *CEN TC251* to define a rigorous and stable information architecture for communicating part or all of the *Electronic Health Record* (*EHR*) of a single subject of care (patient). This is to support the interoperability of systems and components that need to communicate (access, transfer, add or modify) *EHR* data via electronic messages or as distributed objects:

- preserving the original clinical meaning intended by the author;
- reflecting the confidentiality of that data as intended by the author and patient. .

Enabled application

A software application designed to support the use of *SNOMED CT*.

Alternatives

SNOMED CT enabled application
SNOMED enabled application
SNOMED CT application
SNOMED application

Enabled implementation

Implementation of information systems that are able to make effective use of *SNOMED CT* in an organization or region.

Note: *SNOMED CT enabled implementation* has a broader meaning than *SNOMED CT enabled application*. An implementation involves practical deployment of one or more applications but extends beyond the software itself to address personnel and organizational issues that allow the potential benefits to be realized.

Alternatives

SNOMED CT enabled implementation
SNOMED enabled implementation
SNOMED CT implementation
SNOMED implementation

Equivalence

See *Word Equivalents*, *Phrase equivalence* and *Concept equivalence*.

Expression

A structured combination of one or more *concept identifiers* used to express a clinical idea.

Note:

An *expression* containing a single *concept identifier* is referred to as a *precoordinated expression*. An *expression* that contains two or more *concept identifiers* is a *postcoordinated expression*.

The *concept identifiers* in a *postcoordinated expression* are related to one another in accordance with rules expressed in the *SNOMED CT Concept Model*.

These rules allow an *expression* to *refine* the meaning of a *concept* by applying more specific values to particular attributes of a more general *concept*.

Example:

284196006 | burn of skin | : 363698007 | finding site | = 33712006 | skin of hand |

Alternatives

SNOMED CT expression

Expression refinement

The part of a *SNOMED CT expression* that applies qualifying details to a *focus concept*.

Example: A "spiral fracture of the left humerus" can be represented by an *expression* in which the *concept* "fracture of humerus" if made more specific by the addition of two refinements "laterality: left" and "associated morphology: spiral fracture".

Alternatives

Refinement

Extension namespace identifier

See *namespace identifier*.

F

Focus concept

The part of a *SNOMED CT expression* that represents a clinical finding, observation, event or procedure. This *focus concept* may be given context by a surrounding content wrapped and may be made more specific by a refinement.

Example: A past history of replacement of the left hip may be represented by a *SNOMED CT expression* in which the *focus concept* "hip replacement" is refined by "laterality: left" and enclosed in a *context wrapper* representing "past history".

Full release

A *Release Type* in which the *release files* contain every version of every component ever released.

Full view

A view of *SNOMED CT* that includes all the components in a *Full release*. This includes the full history or all components ever released. A *Full view* can be filtered to provide a *Dynamic snapshot view* of the components as they were at any point in the past.

Fully Specified Name

A *term* unique among *active descriptions* in *SNOMED CT* that names the meaning of a *concept* code in a manner that is intended to be unambiguous and stable across multiple contexts.

Notes:

1. *Fully specified names* are indicated with the typeId 900000000000003001 | Fully specified name |.
2. There may be more than one *active description* with the typeId 900000000000003001 | Fully specified name |. However, only one *fully specified name*should be marked as preferred for use in a given *language* or *dialect* in the relevant *Language reference set*.
3. The US English *fully specified name* is the point of reference for the meaning of all *concepts* in the *SNOMED CT International Edition*. However, where a *concept* is part of an *extension* the *fully specified name* specified in the original language of that *extension* applies.

Alternatives

FSN

H

Health Level 7

A not-for-profit, *ANSI*-accredited standards developing organization dedicated to providing a comprehensive framework and related standards for the exchange, integration, sharing, and retrieval of electronic health information that supports clinical practice and the management, delivery and evaluation of health services.

Alternatives

HL7

Health Level 7 Version 3

A standard for communication of health care information developed by *HL7*. Version 3 is based on a formal development framework and its communication structures a derived as refinements from a *Reference Information Model* (*HL7 V3 RIM*).

Alternatives

HL7 V3

Health Level 7 Version 3 Reference Information Model

The *reference information model* on which *HL7 Version 3* is based.

Alternatives

HL7 V3 RIM

Hierarchy

An ordered organization of *concept* codes linked together through |is a| *relationships*. *Concept* codes linked to their more general parent *concept* codes directly above them in a *hierarchy Concept* codes with more general meanings are usually presented as being at the top of the *hierarchy* and then at each level down the *hierarchy* code meanings become increasingly more specific or specialized . Formally, a *hierarchy* is represented as a *Directed Acyclic Graph*.

HL7 TermInfo

An *HL7* project that developed the ' *HL7* Version 3 Implementation Guide: Using *SNOMED CT*' as a *Draft Standard for Trial Use* (DSTU). The purpose of this guide is to ensure that *HL7* Version 3 standards achieve their stated goal of semantic interoperability when used to communicate clinical information that is represented using *concepts* from *SNOMED CT*

Alternatives

Term Info

I

ICD-10

The International Statistical Classification of Diseases and Related Health Problems 10th Revision (ICD-10) is a coding of diseases and signs, symptoms, abnormal findings, complaints, social circumstances and external causes of injury or diseases, as classified by the *World Health Organization*. (*WHO*).

ICD-9

The International Statistical Classification of Diseases and Related Health Problems 9th Revision (ICD-9) is a coding of diseases and signs, symptoms, abnormal findings, complaints, social circumstances and external causes of injury or diseases, as classified by the *World Health Organization*. (*WHO*).

Note: Replaced by *ICD-10*.

ICD-9-CM

The International Classification of Diseases, 9th Revision, Clinical Modification" (ICD-9-CM), Sixth Edition, issued for use beginning October 1, 2008 for federal fiscal year 2009 (FY09). The ICD-9-CM is maintained jointly by the National Center for Health Statistics (NCHS) and the Centers for Medicare & Medicaid Services (CMS).

IFCC IUPAC

Nomenclature, Properties and Units (*C-NPU*) in collaboration with International *Union* of Pure and Applied Chemistry (IUPAC) The *IFCC-IUPAC* coding system Provides a terminology for Properties and Units in the Clinical Laboratory Sciences

Inactive component

A *SNOMED CT component* that is not intended for use. *Active* and *Inactive components* are included in *release files* to provide a historical record of the content of the terminology different points in time.

Note: A component is inactive when the most recent row with the relevant Component.*id* in the *Full Release* of the relevant *Release File* has the value Component.*active*=0 (zero). The most recent row for a component is determined based on the Component. *effectiveTime* value.

Alternatives

Inactive

Inactive concept

A *Concept* that is not intended for use. *Release files* contain *Active* and *Inactive components* to provide a historical record of the content of the terminology at different points in time.

Note: A component is inactive when the most recent row with the relevant Component.*id* in the *Full Release* of the relevant *Release File* has the value Component.*active*=0 (one). The most recent row for a component is determined based on the Component.*effectiveTime* value.

Inactive description

A *Description* that is not intended for use. *Release files* contain *Active* and *Inactive components* to provide a historical record of the content of the terminology at different points in time.

Note: A component is inactive when the most recent row with the relevant Component.*id* in the *Full Release* of the relevant *Release File* has the value Component.*active*=0 (one). The most recent row for a component is determined based on the Component.*effectiveTime* value.

Intellectual property rights

As defined in the IHTSDO affiliate License Agreement: patents, trade marks, service marks, copyright (including rights in computer software), moral rights, database rights, rights in designs, trade secrets, know-how and other *intellectual property rights*, in each case whether registered or unregistered and including applications for registration, and all rights or forms of protection having equivalent or similar effect in any jurisdiction.

Note: The *IHTSDO* owns the *intellectual property rights* of *SNOMED CT*. The *IHTSDO* is responsible for ongoing maintenance, development, quality assurance, and distribution of *SNOMED CT*.

Alternatives

IPR
Intellectual Property
IP

International Health Terminology Standards Development Organisation

The *International Health Terminology Standards Development Organisation* (*IHTSDO*) is a not-for-profit association that develops and promotes use of *SNOMED CT* to support safe and effective health information exchange.

Alternatives

IHTSDO

Intersection

In set theory the *intersection* of the sets A and B, is the set of all objects that are members of both A and B.

Note: Set theory is applied when describing the intended result of combinations of Reference Sets or Constraints.

IS A

The RelationshipType that defines a supertype - *subtype*. *Relationship* between two *Concepts*. Usually expressed as *subtype* | is a | supertype. For Example, Blister with infection | is a | Infection of skin.

ISO

ISO (International Organization for Standardization) is the world's largest developer and publisher of International Standards. ISO is a network of the national standards institutes from over 160 countries, one member per country, with a Central Secretariat in Geneva, Switzerland, that coordinates the system.

ISO TC215

ISO TC215 is the ISO Technical Committee for Standardization in the field of information for health, and Health Information and Communications Technology (ICT). Its objectives are to enable compatibility and interoperability between independent systems, to ensure compatibility of data for comparative statistical purposes (e.g. classifications), and to reduce duplication of effort and redundancies.

K

Kind of value

The nature of a value that may be associated with a *Concept*. For example, the *concept* | systolic blood pressure | can label a numeric value. The Kind-of-Value that it labels is a pressure.

L

Language

For purposes of *SNOMED CT* translations, a *language* is a vocabulary and grammatical form that has been allocated an ISO639-1 *language* code. See also *dialect*.

LOINC

Logical Observation Identifiers Names and Codes, a dataset of universal identifiers for identifying medical laboratory observations and other clinical observations to facilitate exchange and storage of clinical results or vital signs.

Alternatives

Logical Observation Identifiers Names and Codes

M

Machine readable concept model

A representation of the rules that comprise the *SNOMED CT Concept Model* in a form that can be processed by computer software and applied to validate content.

Note: The *Machine readable concept model* can be applied to support consistent authoring of *SNOMED CT* content and can also support the creation of valid *postcoordinated expressions* in instance data.

Alternatives

MRCM

Managed content addition

An implementation strategy that involves creating additional *concepts*, *Descriptions* and *Relationships* in an extension so that data can be recorded to the required level of detail using only *precoordinated expressions*.

Note: A *description logic classifier* can be used to obtain an updated inferred view of the whole terminology in order to support data retrieval.

Alternatives

MCA

Mapping

The process of converting data from a representation in one code system, classification or terminology so that it is represented in another code system, classification or terminology.

Note:

The process as a whole includes the preparation and maintenance of resources used to enable this conversion and the application of such resources to convert instance data.

In *SNOMED CT Mapping* resources are distributed as [see *Simple*] and [see *Complex and Extended Map Reference Sets*]

.

Alternatives

Cross Mapping

Member

A Member of the *International Health Terminology Standards Development Organisation* (*IHTSDO*) in accordance with the *IHTSDO* Articles of Association.

Alternatives

IHTSDO member

Member territory

A territory that is represented by an *IHTSDO Member* (as published by the Licensor from time to time)

Metadata

SNOMED CT content (including *concepts*, *Descriptions* and *Relationships*) that is used to describe or provide additional information about *SNOMED* content and derivatives (including *reference sets*).

Note:

All *SNOMED CT* metadata *concepts* are *subtypes* of 900000000000441003 | SNOMED CT Model Component (metadata) |. The top level of the metadata hierarchy represent broad groups of metadata as shown below.

- 900000000000441003 | SNOMED CT Model Component (metadata) |
  - 106237007 | Linkage concept (linkage concept) | ...
  - 370136006 | Namespace concept (namespace concept) | ...
  - 900000000000442005 | Core metadata concept (core metadata concept) | ...
  - 900000000000454005 | Foundation metadata concept (foundation metadata concept) | ...

Figure 56: Top level of the SNOMED CT metadata hierarchy

Alternatives

SNOMED CT Metadata

Migration

See *Operational migration*, *Data migration* and *Predicate migration*.

Model of meaning

An information model that is structured in a way that is designed to provide a common representation of particular types of information which is reusable between different use cases. A model of a meaning combines structural and terminological component in ways that avoid ambiguity and minimize alternative representations of similar meanings.

Example: A model that specifies a how *SNOMED CT expressions* are used to represent in a particular *reference information model* to represent clinical findings and procedures in an *electronic health record*.

Note: In contrast, a *model of use* represents the underlying meaning in a way that is determined by a limited set use cases.

Model of use

An information model that is structured in a way suggested by a particular intended use of the information that will be represented by that model.

Example: A database that is structured with tables and fields that match specific *user interface* forms and the data entry box on those forms.

Note: In contrast, a *model of meaning* represents the underlying meaning in a way that is common to and reusable between different use cases.

Modeler

A person who directly edits the logic definitions and other structures of the terminology. Also sometimes called Clinical Editor or Terminology Manager.

Alternatives

SNOMED CT modeler
Modeller
SNOMED CT author

Modeling

The process of editing logic definitions to reflect the meaning intended by the *Fully Specified Name*.

Alternatives

SNOMED CT modeling
Modelling
SNOMED CT authoring

Monohierarchy

A *Monohierarchy* is a hierarchy in which each node is linked to one and only one parent node.

This type of hierarchy can be represented as a tree with a single root to which each node is attached.

Alternatives

Monohierarchical classification

Moved elsewhere

A *Status* value applicable to a *component* that has been moved to another *Namespace*. *Concepts* or *Descriptions* may be moved from an *Extension* to the *International Release*, from the *International Release* to an *Extension* or between one *Extension* and another. Moves occur if responsibility for supporting the *Concepts* changes to another organization .

Note: Component status value

N

Namespace concept

A *Concept* that exists to represent a *SNOMED CT Namespace-Identifier*. All *Namespace Concepts* are direct *subtypes* of the *Concept* "*Namespace Concept* which is a *subtype* of the Top-Level *Concept* "Special *Concept* ".

Namespace identifier

A seven digit number allocated by the *IHTSDO* to an organization that is permitted to maintain a *SNOMED CT Extension*. The *namespace identifier* forms part of the *SCTID* allocated every *component* that originated as part of an *Extension*. Therefore, it prevents collision between *SCTIDs* issued by different organizations . The *namespace-identifier* indicates the provenance of each *SNOMED CT component*.

Note: Short format *SCTIDs*,which are used for *components* that originate in the *International Release*, do not include a *namespace-identifier*. In this case the *partition identifier* provides sufficient information about the origin of the component.

Alternatives

Extension namespace identifiers
NamespaceId

National Health Service

Located in the United Kingdom, the *National Health Service* (*NHS* worked with the College of American Pathologists in the development of *SNOMED CT*. The *NHS* is was one of the founder Members of the *IHTSDO* that is now responsible for *SNOMED CT*.

Alternatives

UK National Health Service
UK NHS
NHS

National Library of Medicine

The *National Library of Medicine* (*NLM*, in Bethesda, Maryland, is a part of the National Institutes of Health, US Department of Health and Human Services (HHS). *NLM* is the world's largest medical library. The *NLM* represents the US, as a founder Member of the *IHTSDO*.

Alternatives

NLM

National Release Center

The organization within an *IHTSDO Member* country that is responsible for maintaining and releasing *SNOMED CT* content including any National *Extensions* of *SNOMED CT*.

Natural language processing

Natural *Language* processing (*NLP* is concerned with the interactions between computers and human-readable *languages*. *NLP* includes understanding and generation of human-readable representations. *NLP* understanding systems convert human-readable text into formal representations, which may for example include *SNOMED CT expressions*, to enable more effective processing by other software. *NLP* generation systems convert information from formal representations into human-readable text.

Alternatives

NLP

Navigation

The process of locating a *Concept* by traversing *Relationships* or *Navigation links*. For example, moving from a supertype *Concept* to more refined *Concepts*, from a specific *Concept* to a more general *Concept* or from a *Concept* to its *Defining characteristics*. *Navigation Links* allow *navigation* to follow intuitive routes through *SNOMED CT* even where there are no direct supertype or *subtype Relationships*.

Navigation concept

A *Concept* that exists only to support *Navigation*. A *Navigation Concept* is not suitable for recording or aggregating information. All *Navigation Concepts*:

- Are direct *subtypes* of the *concept* "Navigational *Concept*";
- Have not other supertype or *subtype Relationships*
- Are linked to other *Concepts* only by Navigational Links.

Navigation Hierarchy

A hierarchical view of a set of *SNOMED CT concepts* that is intended to assist navigation at the *user interface*.

Note: There are several differences between *navigation hierarchies* and the formal *subtype hierarchy*:

1. Links between *concepts* in a *navigation hierarchy* are represented by an [see *Ordered Reference Set*]
2. *Navigation links* do not contribute to the semantic definitions of *concepts*. Therefore, the criteria for creating a *navigation hierarchy* can be based on arbitrary criteria relating to usability;
3. A *navigation hierarchy* may specify the order in which a set of *concepts* are to be displayed when nested under another specified *concept*.

Non-member territory

A territory that is not an *IHTSDO Member* Territory

Note: In accordance with *IHTSDO affiliate* License, fees are payable to the *IHTSDO* for use of *SNOMED CT* in non-Member Territories.

Normal form

A representation of a *SNOMED CT expression* in which none of the referenced *concepts* are *fully defined* and where there is no redundancy or duplication of meaning.

Notes:

1. *Normal forms* can be used to determine *equivalence* and subsumption between *expressions* and thus assist with selective retrieval.
2. Any *SNOMED CT expression* can be transformed to its *normal form* by replacing each reference to a *fully defined concept* with a nested *expression* representing the definition of that *concept*. Transformation rules then resolve redundancies, which may arise from expanding *fully defined concepts*, by removing less specific *attribute values*.

Normal form transformation

The process of converting a *SNOMED CT expression* into its *normal form*.

Notes:

1. The *normal form* provides a way compare different *expressions* which have a similar meaning.
2. The transformation rules are described in [see *Transforming expressions to normal forms*].

Alternatives

Transform
Transformation

O

openEHR

openEHR is an international not-for-profit Foundation working toward making the interoperable, life-long *electronic health record* a reality and improving health care in the information society. It is develops specifications that are primarily based on and extend key aspects of the *CEN* Standard for *Electronic Health Record* Communication (EN 13606).

Operational migration

Steps taken to enable an organization that either used a previous coding scheme (or no clinical coding scheme) to make use of *SNOMED CT*.

P

Partition-identifier

The second and third digits from the right of the string rendering of the *SCTID*. The value of the *partition-identifier* indicates the type of component that the *SCTID* identifies (e.g. *Concept*, *Description*, *Relationship*, etc) and also indicates whether the *SCTID* contains a *namespace identifier*.

Alternatives

PartitionId

Pending move

A *Status* value applicable to a *component* that is thought to belong in a different *Namespace* but which is maintained with its current *SCTID* while awaiting addition to the new *Namespace*. A new *Concept* and associated *Descriptions* may be added with this *Status* where a missing *SNOMED CT Concept* is urgently required to support the needs of a particular *Extension*. Existing *Concepts* are also given this *status* when it is recognized that they should be moved to a different *Extension* or to the *International Release*. See also *Moved elsewhere*.

Note: Component status value.

Phrase equivalence

Two words or phrases with a similar meaning. For example, "renal calculus" and "kidney stone". See *Word Equivalents*.

Polyhierarchy

A *Polyhierarchy* is a hierarchy in which each node has one or more parents.

This type of hierarchy can be represented as a graph in which each node has a one or more directed links to or from other nodes. Since a node in a hierarchy cannot be a *descendant* of itself the resulting graph must not contain cyclic *Relationships*. This type of graphs is referred to as a "*Directed Acyclic Graph*".

Alternatives

Polyhierarchical classification

Postcoordinated expression

Representation of a clinical meaning using a combination of two or more *concept identifiers* is referred to as *postcoordination*.

Note: Some clinical meanings may be represented in several different ways. *SNOMED CT* technical specifications include guidance for transforming logical *expressions* to a common *canonical form*.

Example: *SNOMED CT* includes the following *concepts*:

125605004 | fracture of bone |
363698007 | finding site |
71341001 | bone structure of femur |

SNOMED CT also includes a *precoordinated concept* for 71620000 | fracture of femur |. Therefore It is possible to represent the clinical meaning "fracture of femur" in different ways:

- as a *precoordinated expression*:
  - 71620000 | fracture of femur |
- or as a *postcoordinated expression*:
  - 125605004 | fracture of bone | : 363698007 | finding site | = 71341001 | bone structure of femur |

Alternatives

Postcoordinated
Postcoordination

Precoordinated expression

Representation of a clinical meaning using a single *concept identifier* is referred to as a *precoordinated expression*.

Note: In contrast, *expressions* that contain two or more *concepts Identifier* are referred to as *postcoordinated expressions*. For more information and examples see the glossary entry for *postcoordinated expression*.

Alternatives

precoordinated expression
Precoordinated
Precoordination

Predicate migration

Steps taken to enable pre-existing data retrieval predicates (including queries, standard reports and decision support protocols) to be converted or utilized in a system using *SNOMED CT*.

Preferred term

The *term* that is deemed to be the most clinically appropriate way of expressing a *Concept* in a clinical record. The *Preferred Term* varies according to language and *dialect*.

Note: In *Release Format 2* the *Preferred Term* is indicated by the *acceptabilityId* field of a *Language Refset*.

Note: In *Release Format 1* the *Preferred Term* is indicated by a *Language Subset* and/or the *DescriptionType* field of the *Description file*.

Primitive concept

A *concept* with a formal logic definition that is not sufficient to distinguish its meaning from other similar *concepts*.

Note:

The meaning of *SNOMED CT concept* is expressed in a human-readable form by its *Fully Specified Name*. Each *concept* also has a formal logic definition represented by a set of defining *relationships* to other *concepts*. This logic definition is computer processable. A *primitive concept* does not have sufficient defining *relationships* to computably distinguish them from more general *concepts* (supertypes).

See also *sufficiently defined concept*.

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".

- 116680003 | is a | = 74400008 | appendicitis |
- 116676008 | associated morphology | = 23583003 | inflammation |
- 363698007 | finding site | = 66754008 | appendix structure |

Figure 57: Definition of: |atypical appendicitis (disorder)| (primitive)

Q

Qualifying characteristic

An *attribute-value relationship* associated with a *concept* code to indicate to users that it may be applied to refine the meaning of the code. The set of qualifying *relationships* provide syntactically correct values that can be presented to a user for *postcoordination*. Example: 'Revision *status* ' = 'First revision' is a possible *qualifying characteristic* of 'Hip replacement'. A *qualifying characteristic* is contrasted with a *defining characteristic*. It is referred to in *CTV3* as a '*Qualifier*.

Alternatives

Qualifier

Quality characteristic

A type of attribute of a component by which its quality is assessed or measured.

Note: The set of *IHTSDO quality characteristics* are a typology of attributes of an *IHTSDO* Component by which its quality is assessed or measured. A typology is the study or systematic classification of types that have attributes or traits in common.

Quality metric

An agreed method and means for measuring levels of achievement, performance or conformance of a component or its *Quality characteristic*(s).

Quality target

An agreed level of achievement, performance or conformance of a component for any given *Quality characteristic*.

Query predicate

A statement of a condition that determines whether candidate instance data should be included in or excluded from a selection.

Note: *Query predicates* applied to a set of *SNOMED CT expressions* may test for subsumption of the overall meaning and/or may test the values applied to particular *attributes* in the *expression*.

R

Range

A constrained set of values that the *Concept Model* permits to be applied to a specific *attribute* when that *attribute* is applied to a *concept* in a particular *domain*.

Note: The *range* of permitted values that can be applied to an *attribute* is typically defined to include concepts in one or more branches of the subtype hierarchy.

Example: The *range* for values of 116676008 | Associated morphology | is subtypes of 49755003 | Morphologically abnormal structure |.

Alternatives

Concept model range

Read Code

A five-character code allocated to a *concept* or *term* in *CTV3*. Note that codes allocated in *Read Codes Version 2* and the *Read Codes 4-Byte Set* are also included in *CTV3*. The original 4-byte codes are distinguished from 5-byte codes in the general representation by prefixing them with a full stop.

Alternatives

Read Codes 4-Byte Set
Read Codes Version 2

Realm

A sphere of authority, expertise, or preference that influences the range of *components* required, or the frequency with which they are used. A *Realm* may be a nation, an organization , a professional discipline, a specialty, or an individual user.

Record services

Functions performed by software that interacts with a record system used to capture information which may include references to information in a terminology.

Note: *Record services* are intimately related to ways in which information is entered, stored and retrieved by a particular application. These services interact with *Terminology services* but, unlike *Terminology services* they are usually specific to a particular application.

Reference information model

A high-level generalized model that allows information to be represented and related consistently within a particular field of human endeavor.

Note: The *Health Level 7 Version 3 Reference Information Model* is the most widely used *reference information model* in health care.

Reference set

A work consisting of a set of references to *SNOMED CT components* that may associate additional properties with *components* that are members of the set and/or which may indicate associations between members of the set or between members of the set and content of another nomenclature, classification or knowledge structure. The uses of *Reference sets* include identification of subsets of *SNOMED CT* content, representation of alternative hierarchical structures and *maps* to classifications.

Alternatives

SNOMED CT reference set
Refset

Reference set member

A uniquely identified row within the *snapshot view* of a *reference set*.

Note:

1. Different versions of a *reference set member* may share the same identifier (*id*) but have different *effectiveTimes*. This allows a *reference set member* to be modified or made *inactive* (i.e. removed from the active set) at a specified time.
2. Each *reference set* has an identifier (*refsetId*) and contains one or more *reference set members*. Each *reference set member* has its own unique identifier (*id*) which allows it to be versioned using the *effectiveTime* and *active* fields. All *reference set members* also contain a *referencedComponentId* (which refers to a component that is part of the set) and other fields that depend on the type of *reference set*.

Reference terminology

A terminology in which each *term* has a formal computer processable definition that supports meaning based retrieval and aggregation. *SNOMED CT* is a *reference terminology*

Relationship

An association between a source *concept* and a destination *concept*. The nature of the association is indicated by a reference to another *concept* referred to as the *relationship type*.

Notes:

1. Each *relationship* provides information about the source *concept*. In the example below
2. *Relationships* are represented by rows in the *Relationship File*

Example:

Table 262: Illustrative example of a *Relationship*

source	type	destination
74400008 | appendicitis |	363698007 | finding site |	66754008 | appendix structure |

Alternatives

SNOMED CT relationship

SNOMED CT Release

The content of a version of a *SNOMED CT Edition* that has been made available to licensees at a particular point in time.

Release file

A computer file used to distribute *SNOMED CT* content from the *IHTSDO* (or from the originator of an *Extension*) in a form that can be readily imported by a software application.

SNOMED CT release files follow one of the *release format* specifications *RF1* or *RF2*.

Alternatives

SNOMED CT release file
SNOMED CT distribution file

Release format

A file structure specified by the *IHTSDO* for files used to distribute *SNOMED CT* content.

Note: There are currently two *release formats*: *Release Format 1* and *Release Format 2*.

Alternatives

SNOMED CT release format
SNOMED CT distribution format

Release Format 1

The file structure specified by the *IHTSDO* for the files used to distribute *SNOMED CT* content in 2002.

Note: This format was replace by *Release Format 2* in January 2012, which is now the primary format for the *SNOMED CT International Release*. However, for backward compatibility *Release Format 1* files can be generated using a conversion utility and continue to be distributed available during an interim transitional period.

Alternatives

SNOMED CT Release Format 1
RF1

Release Format 2

The file structure specified by the *IHTSDO* for files used to distribute *SNOMED CT* content from 2011.

Note: See also: *Release Format 1*.

Alternatives

SNOMED CT Release Format 2
RF2

Release Type

The temporal scope and completeness of a *Release Format 2* file or set of files.

Table 263: SNOMED CT Release Types

Release Type	***Description***
Full	The files representing each type of component contain **every** version of every component ever released.
Snapshot	The files representing each type of component contain **one** version of every component released up to the time of the snapshot. The version of each component contained in a snapshot is the most recent version of that component at the time of the snapshot.

Release Type	***Description***
Delta	The files representing each type of component contain only component versions created since the previous release. Each component version in a *delta release* represents either a new component or a change to an existing component.

Root concept

The single *concept* that is at the top of the | SNOMED CT Concept | hierarchy.

Root metadata concept

The single *concept* that is at the top of the | SNOMED CT Model Component (metadata) | hierarchy.

Note: Most of the data in the metadata hierarchy is only relevant to *Release Format 2*. Therefore, this *concept* may not be present in some *Release Format 1* files.

Alternatives

Root metadata code

S

Situation with explicit context

A *concept* that specifically includes a definition the context of use of a clinical finding or procedure.

Example: "Family history of diabetes mellitus" is a situation with explicit *concept* because it defines the context as "family history". In contrast, "diabetes mellitus" is not a *situation with explicit context* because it can be used in many different situations including "family history", "past medical history", "current diagnosis", etc.

Note: A *situation with explicit context* is defined as a *subtype* of the situation to which it applies with an attribute associating it with the relevant clinical finding or procedure.

Alternatives

Explicit context
Clinical situation

Snapshot release

A *Release Type* in which the *release files* contain one version of every component released up to the time of the snapshot. The version of each component contained in a snapshot is the most recent version of that component at the time of the snapshot.

Snapshot view

A view of *SNOMED CT* that includes all the components in the state there were in at a specified point in time. A *Snapshot view* be provided by a fixed representation that matches the content of a *Snapshot release* or may be generated as a *Dynamic snapshot view* by filtering a *Full view*.

SNOMED

An acronym for the **S**ystematized**No**menclature of **Med**icine originally developed by the College of American Pathologists and now owned and maintained by the *IHTSDO*. *SNOMED Clinical Terms* is the most recent version of this terminology. It was preceded by *SNOMED RT* and *SNOMED International*.

SNOMED Clinical Terms

SNOMED CT is a clinical terminology maintained and distributed by the *IHTSDO*. It is considered to be the most comprehensive, multilingual healthcare terminology in the world. It was created as a result of the merger of *SNOMED RT* and *NHS Clinical Terms Version 3*.

Alternatives

SNOMED CT

SNOMED CT Edition

The combination of a *SNOMED CT Extension* with the *SNOMED CT International Edition* and, where relevant, any module from other *Extensions* on which the *SNOMED CT Extension* depends.

Note: A *SNOMED CT Edition* may be released by the provider of the *SNOMED CT Extension*. However, in general a *SNOMED CT Edition* is derived by combining the *SNOMED CT Extension* release files with relevant release data from the *SNOMED CT International Edition* and any other *Extensions* on which it depends.

Alternatives

Edition

SNOMED CT Extension

A set of terminology *components* and *derivatives* that add to and are dependent on the *SNOMED CT International Edition*, and are created, structured, maintained and distributed in accordance with *SNOMED CT* specifications and guidelines.

Notes:

1. *Components* that are created in an *extension* are identified using *extension SCTIDs*. These identifiers include an *extension namespace* which ensures that they do not collide with other *SCTIDs*, and can be traced to an authorized originator.
2. *Namespace identifiers* are allocated in response to requests from *IHTSDO Members* and *Affiliates*. For further information about this process and for access to the current SNOMED CT Namespace Register please refer to the *IHTSDO web page on Namespaces*.
3. *IHTSDO Members* may create, maintain and distribute *extensions* to address specific national, regional and language requirements. *IHTSDO Affiliates* may also create, maintain and distribute *extensions* to meet the needs of particular software solutions and customers.
4. See also *Edition* which refers to the combination of an *extension* with the *International Release* and, where relevant, any modules from other *extensions*on which it depends.

Alternatives

Extension

SNOMED CT Identifier

A unique *integer* identifier applied to each *SNOMED CT component* (*Concept*, *Description*, *Relationship*, *Subset*, etc.). Each *SCTID* includes an item identifier, a *check-digit*, a *partition identifier* and, depending on the *partition identifier*, may also include a *namespace identifier*.

Alternatives

Identifier
SCTID

SNOMED CT International Edition

The part of *SNOMED CT* that is maintained and distributed by the *IHTSDO* and available to all *IHTSDO Members* and *Affiliates* as the shared foundation of the terminology.

Notes:

1. The *International edition*, provided by the *IHTSDO*, may be supplemented by *Extensions* maintained by *IHTSDO Members* and *Affiliates* to meet additional national, local and organizational requirements.
2. The combination of the *International edition* with a *National Extension* is referred to as a *National Edition*.
3. The *International release* refers to a release of content from the *International edition* at a particular release date.

Alternatives

International edition

SNOMED CT International Release

The set of *release files* provided on a specified release date, to represent the part of the content of *SNOMED CT* that forms the common foundation to the terminology available to all *IHTSDO Members* and *Affiliates*.

Notes:

1. The *International release*, provided by the *IHTSDO*, may be supplemented by *Extension* releases provided by *IHTSDO Members* and *Affiliates* to meet additional national, local and organizational requirements.
2. See also *International Edition* which refers to the same general content, without specifying a particular release date.

Alternatives

International Release

SNOMED CT Module

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

Notes:

1. *Components* and *reference set members* that are part of the same *module* are share the same *moduleId* value.
2. Each *component* and *reference set member* is a part of one and only one *module* as at a given point in time.
3. The organization responsible for a *module* can move a *component* and *reference set member* from that *module* to another *module* that the same organization is responsible for, by creating a revised version of the *component* or *reference set member* with a different *moduleId* that applies from the *effectiveTime* of the revised version.
4. Subject to rules related to movement of components between two extensions or between an extension and the International Edition, it is possible for a *component* and *reference set member* to be moved between *modules* maintained by different organizations.

Alternatives

Module

SNOMED CT National Edition

The combination of a *National Extension* with the *SNOMED CT International Edition* and, where relevant, any module from other *Extensions* on which the *National Extension* depends.

Note: The *National Edition* may be made available to licensees at a particular release date as part of a *National Release*. However a *National Edition* can also be derived by combining the *National Extension* release files with relevant release data from the *SNOMED CT International Edition* and any other *Extensions* on which it depends.

Alternatives

National Edition

SNOMED CT National Extension

A *SNOMED CT Extension* that is maintained by an *IHTSDO Member* for use in a particular country.

Note: See also *National Edition* which refers to the combination of a *National Extension* with the *International Release* and, where relevant, any modules from other *Extensions*on which it depends.

Alternatives

National Edition

SNOMED CT National Release

A National Extension and/or *National Edition* as made available to licensees by an *IHTSDO Member* at a particular release date.

Notes:

1. The *National Release* is made available as a set of *release files* which contain components and derivatives from a National Extension maintained and distributed by an *IHTSDO Member*.
2. A *National release* may also include the *SNOMED CT International Release* on which it depends, in which case it is a release of the *National Edition*.
3. Alternatively, a *National Release* may consist only of the *National Extension release files* for the specified release date. In this case, the *National Edition* is generated by combining these files with the *International Release* on which it depends.

Alternatives

National Release

SNOMED International

SNOMED International is the version of *SNOMED* ® that was first released in 1993 and which, as version 3.5 released in 1998, It was the immediate predecessor of *SNOMED RT*.

SNOMED Reference terminology

The version of *SNOMED* ® prior to the collaborative effort to develop *SNOMED Clinical Terms*. It was one of the source terminologies, along with *CTV3*, from which *SNOMED CT* was developed.

Alternatives

SNOMED RT

Sponsored Territory

A Non-Member Territory that has been recognized and designated by the Licensor (*IHTSDO*) as a sponsored territory

Note: *SNOMED CT* may be used free of charge by *IHTSDO affiliates* and their sub-licensees in Sponsored Territories. Information about Sponsored Territories is published on the *IHTSDO* web site.

Stated view

The *stated view* of a *Concept* definition consists of the *Relationships* directly edited by terminology authors. It consists of the stated *subtype Relationships* plus the defining *Relationships* that exist prior to running a *Description Logic classifier*.

Note: The *Relationships* distributed in the main *Relationships* files are inferred from the stated *Relationships* using a *Description Logic classifier* to ensure consistency and completeness. The *stated view* is distributed in the *Stated Relationships File*.

Alternatives

Stated form

Statistical classification

A hierarchical organization of *terms* or ideas that allows aggregation into categories that can be counted and compared without double counting. A *statistical classification* is monohierarchical which means that each node in the *hierarchy* is part of one node is the level above. This avoids double counting but means that arbitrary decisions must be made where a node is naturally related to more than one parent. For example, in a *statistical classification* such as *ICD-10*, 'bacterial pneumonia' is be related to 'lung disorder' or 'infection disorder' but not to both.

Structure-Entire-Part

A modeling approach used in SNOMED CT to represent anatomical entities such as body organs, body systems, body regions,etc.

- **Stucture** is the most general way to refer to an organ, body system or region.
- **Entire** refers to a complete organ, body system or region.
- **Part** refers to a part of an organ, body system or region. It explicitly does not refer to the entire organ, body system or region.

Example: The figure below illustrates the relationships between the structure, entire and part concepts applied to a the heart.

- 80891009| heart structure |
- 302509004| entire heart |
- 119202000| heart part |

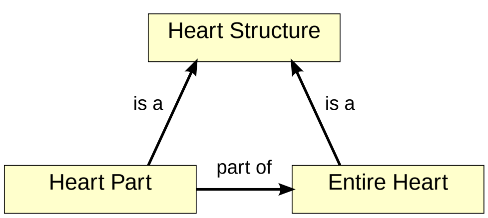


Figure 58: Structure-Entire-Part applied the heart

Alternatives

SEP

Subset

A set of *components* which part of and fully included with a larger set (e.g. a specified set of *Concepts* or *Descriptions*)

Notes:

1. In Release Format 2 the standard way to represent a *subset* of *components* is by using a *Simple Reference Set*
2. In Release Format 1 the term *subset* has the following special meaning:

   - A group of *components* (e.g. *Concepts*, *Descriptions* or *Relationships*) that share a specified common characteristic or common type of characteristic. *Subsets* represent information that affects the way the *components* are displayed or otherwise accessible within a particular *realm*, specialty, application or context.

   This special meaning arose from the "Subset Mechanism" which has now been replaced by *Reference Sets*. Therefore, except when referring to RF1 files the term *subset* should should now be used for its more correct general meeting.

Subsumption test

A test to determine whether a specified candidate *concept* or *expression* is a *subtype descendant* of another specified *concept* or *expression*.

Alternatives

Subtype test

Subtype

A specialization of a *concept*, sharing all the definitional attributes of the parent *concept*, with additional *defining characteristics*. For example, bacterial infectious disease is a *subtype* of infectious disease. Bacterial septicemia, bacteremia, bacterial peritonitis, etc. are *subtypes* of bacterial infectious disease (and infectious disease as well). *Subtype* is sometimes used to refer to the *concepts* in a *hierarchy* that are directly related to a parent *concept* via the |is a| *relationship*. In this usage, it is distinguished from *descendants* which explicitly includes *subtypes* of *subtypes*

Subtype child

A *concept* that has a direct |is a| *subtype Relationship* to a specified *concept*. See also *subtype* and *subtype descendant*.

Example:

The figure below shows an example hierarchy in which *concept* "E" has three *subtype children* (G, H and J).

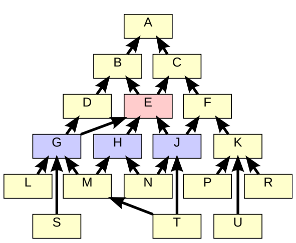


Figure 59: Hierarchy Illustration - Subtype children

Alternatives

Subtype children
Child
Children

Subtype classification

A classification hierarchy in which each node is connected to its supertypes. This allows aggregation of information based on a hierarchy of types.

Alternatives

Subtype hierarchy

Subtype descendant

All *subtypes* of a *concept*, including *subtypes* of *subtypes*. For example, if a *concept* has four *children*, then *descendants* are those *children* plus all the *concepts* that are descended from those four *children*. See also *subtype* and *subtype child*.

Example:

The figure below shows an example hierarchy in which *concept* "E" has eight *subtype descendants* (G, H, J, L, M, N, S and T).

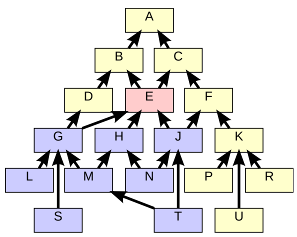


Figure 60: Hierarchy Illustration - Subtype descendants

Alternatives

Descendant

Sufficiently defined concept

A *concept* with a formal logic definition that is sufficient to distinguish its meaning from other similar *concepts*.

Note:

The meaning of *SNOMED CT concept* is expressed in a human-readable form by its *Fully Specified Name* (FSN) and has a formal logic definition represented by a set of defining *relationships* to other *concepts*. A *Sufficiently defined concept* has sufficient defining *relationships* to computably distinguish it from other *concepts*.

See also *primitive concept*.

Example: The *concept* 74400008|appendicitis (disorder)| is *sufficiently defined* by the following definition because any *concept* for which this definition was true would be the disorder "appendicitis".

- 116680003 | is a | = 18526009 | disorder of appendix |
- 116680003 | is a | = 302168000 | inflammation of large intestine |
- 116676008 | associated morphology | = 23583003 | inflammation |
- 363698007 | finding site | = 66754008 | appendix structure |

Figure 61: Definition of: |appendicitis (disorder)| (sufficiently defined)

Alternatives

Fully defined concept

Supertype ancestor

Any *concepts* of which the specified *concept* is a *subtype*. Includes the *supertype parents* and the *supertype parents* of each *supertype parent* and so on recursively until the *root concept* is reached.

Example:

The figure below shows an example hierarchy in which *concept* "T" has ten *supertype ancestors* A, B, C, D, E, F, G, H, J, and M).

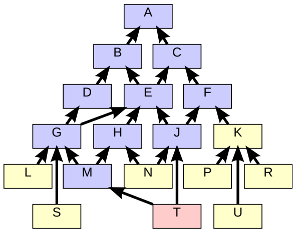


Figure 62: Hierarchy Illustration - Subtype ancestors

Alternatives

Ancestor

Supertype parent

A *concept* that is the target of a direct | is a | *subtype Relationship* from a specified *concept* (see also *supertype ancestor*).

Example:

The figure below shows an example hierarchy in which *concept* "T" has two *supertype parents* (J and M).

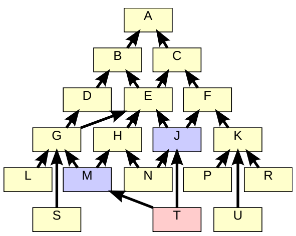


Figure 63: Hierarchy Illustration - Supertype parents

Synonym

A *term* that is an acceptable way to express a the meaning of a *SNOMED CT concept* in a particular *language*.

Note:

1. *Synonyms* are represented as *SNOMED CT descriptions* with the *typeId* value 900000000000013009 | Synonym |.
2. *Synonyms* allow representations of the various ways a *concept* may be described.
3. *Synonyms* (unlike *fully specified names*) are not necessarily unique because the same *term* can be used to describe more than one *concept*.
4. The *preferred term* is the *synonym* marked as preferred for use in the *Language reference set*for a given *language* or *dialect*.

T

Target code

A code or other *Identifier* within a *Target Scheme*.

Target scheme

A terminology, coding scheme or classification to which some or all *SNOMED CT Concepts* are mapped.

Technology Preview

An experimental status applied to a collection of *SNOMED CT release files* that represent a proposed additions of *components* and/or *derivatives* to the *SNOMED CT International Release* or to a *SNOMED CT Extension*. The *Technology Preview* status indicates the releasing party (*IHTSDO* or the owner of the *Extension*) is only releasing these additional *components* or *derivatives* for review and testing by implementers and other stakeholder. The objective of a *Technology Preview* is to test the chosen approach and elicit comments before committing to the content and/or release format for the additional material. It is likely that, prior to release of a *Candidate Baseline*, significant changes may be made to address comments made and issues identified by testing.

Note: The significance a *Technology Preview* release is that this data should not be used in an operational environment that may incorporate the data into a record or create a dependency on continued maintenance of the additional *components* or *derivatives*.

Term

A human-readable phrase that names or describes a *concept*. A *term* is one of the properties of a *description*. Other properties of a *description* link the *term* to an identified *concept* and indicate the type of *description* (e.g. *Fully Specified Name*, *Synonym*, etc.).

Terminology binding

A link between a terminology component and an information model artifact, such as class or attribute in a *electronic health record* or message.

Notes:

1. Terminology components include *SNOMED CT expressions*, *reference sets* and constraints.
2. Information model artifacts include classes and attributes in reference models for *electronic health records* and communication specifications.
3. *Terminology binding* can also be used to refer to the process of creating and persisting links between terminology components and information model artifacts.

Examples:

1. A set of coded values that may be applied to a particular attribute in an information model. The set may be expressed either explicitly (extensionally) or as a definitional constraint (intensionally).
2. The association between a named attribute value in the information model and a specific coded value or *expression*.
3. A rule that determines the way that a coded *expression* is constructed based on multiple attribute values in the information model.

Terminology server

Software that provides access to *SNOMED CT* (and/or to other terminologies). A *terminology server* typically supports searches and *Navigation* through *Concepts*. A server may provide a *user interface* (e.g. a *browser* or set of screen controls) or may provide low-level software services to support access to the terminology by other applications. See the *SNOMED CT Technical Implementation Guide*.

Alternatives

SNOMED CT terminology server

Terminology services

Functions performed by software that interacts with one or more representations of the terminology and provide access to information derived from the terminology.

Note: *Terminology services* can be generalized , so that they are independent of the way the terminology is used in a particular application. *Terminology services* may be used by *record services* that enter, store and retrieve information that includes *SNOMED CT expressions*. In contrast to *terminology services*, *record services* are usually specific to the design of a particular application.

Textual definition

An additional textual *description* applied to some *SNOMED CT concepts* that provides additional information about the intended meaning or usage of the *concept*.

Note:

Textual definitions are distributed in a file that follows the same structure as the *Description file* (RF2) but the terms permitted by the "textual definition" are much longer the 255 character limited applied to *synonyms* and *fully specified names*. Textual definitions are not essential for *SNOMED CT implementations* but they are useful as they provide narrative *Descriptions* of *concepts* that may be easier to understand than the shorter terms.

These *Descriptions* go beyond the detail of the *Fully Specified Name* as shown in the example below.

Example:

Table 264: Textual Definition

conceptId	*Fully Specified Name*	Textual Definition
11530004	Brittle diabetes mellitus (finding)	Diabetes mellitus in which there are frequent, clinically significant fluctuations in blood glucose levels both above and below levels expected to be achieved by available therapies.

Top level concept code

A *Concept* Code that is directly related to the *Root Concept* Code by a single *Relationship* of the *Relationship Type* | is a |. All *Concept* Codes (except for metadata *concepts*) are descended from at least one Top-Level *Concept* Code via at least one series of *Relationships* of the *Relationship Type*"| Is a | ".

Top level metadata code

A *Concept* Code that is directly related to the *Root Metadata Code* by a single *Relationship* of the *Relationship Type* | is a |. All Metadata *Concept* Codes are descended from at least one Top-Level Metadata *Concept* Code via at least one series of *Relationships* of the *Relationship Type*"| Is a | ".

Note: Most of the data in the metadata hierarchy is only relevant to Release Format 2. Therefore, this concept may not be present in Release Format 1 files.

Transitive closure

A comprehensive view of all the *supertype ancestors* of a *concept* derived by traversing all the | is a | *relationships* between that *concept* and the *root concept*.

Note: A *transitive closure* table represents the *transitive closure* of all *active concepts*.

Translation

The process of rendering text originally written in one language (source language) into another language (target language).

Translation source language

The language in which the original text is written.

Example: English is the source language for the *International edition* of *SNOMED CT*.

Alternatives

Source language

Translation target language

A language into which the original text is being translated or rendered.

Example: For the Spanish language edition, Spanish is the target language.

Alternatives

Target language

Translation Service Provider

Person or organization supplying a translation service.

Alternatives

TSP

U

Understandability, Reproducibility and Usefulness

Criteria applied to test the validity of new *concepts* and design features of *SNOMED CT*.

- Understandable: The meaning of a *concept* can be understood by an average health care provider, without reference to private or inaccessible information.
- Reproducible: Multiple users apply the *concept* to the same situations.
- Useful: The *concept* has a practical value to users that is self-evident or can be readily explained.

Alternatives

URU

Union

In set theory *union* of the sets A and B, is the set of all objects that are a member of A, or B, or both.

Note: Set theory is applied when describing the intended result of combinations of Reference Sets or Constraints.

User interface

The way a software application presents itself to a user including, its on screen appearance, the commands it puts at a users disposal, and the manner in which the user can access and update information by using the application.

Alternatives

UI

V

Value Set

A uniquely identifiable set of valid concept representations, where any concept representation can be tested to determine whether or not it is a member of the *value set*.

Notes:

1. This definition is used in *HL7* Vocabulary Working Group documents. In *SNOMED CT* a concept representation may be a *concept* identifier or a *SNOMED CT Expression*.
2. A *Reference set* can be used to represent a value set of *SNOMED CT concepts* each of which is represented by a *concept* identifier in the *referencedComponentId* field.

W

Word equivalent

A word or abbreviation that is stated to be equivalent to one or more other words, phrases or abbreviations for the purposes of textual searches of *SNOMED CT*. *Word Equivalents* and Phrase equivalents are represented as rows in the *Word Equivalents Table*.

Workbench

A set of *IHTSDO* sponsored software tools designed to support the development, maintenance, and use of *SNOMED CT* in health systems around the world.

Alternatives

IHTSDO Workbench

World Health Organization

the directing and coordinating authority for health within the United Nations system. The *World Health Organization* (*WHO* maintains the International *Statistical Classification* of Diseases and Related Health Problems (ICD).

Alternatives

WHO