INTERNATIONAL HEALTH TERMINOLOGY
STANDARDS DEVELOPMENT ORGANISATION

SNOMED CT
Release Format 2.0
Reference Set Specifications

Date 20100706
Version 1.0

Amendment History

Version	Date	Editor	Comments
0.1	20090610	John Gutai (IHTSDO)	Draft for internal comment
0.2	20090605	John Gutai (IHTSDO)	First draft for review by Tech Committee
0.3	20090630	John Gutai (IHTSDO)	Inclusion of support for MRCM
0.4	20090709	John Gutai (IHTSDO)	Updated following review
0.5	20090724	John Gutai	Included comments from Community of Practice Review
0.6	20090821	John Gutai	Updated to DFTU with comments from the Committee and Member reviews and QA review.
0.8	20100330	John Gutai	Updated with comments following Draft for Trial Use period.
0.9	20100521	John Gutai	Updated following post DFTU review
0.10	20100521	John Gutai	Updated following final QA review
0.11	20100529	John Gutai	Final feedback from the Technical Committee and UKTC
1.0	20100706	John Gutai	Approved by Management Board as an IHTSDO Standard

Associated Documentation

Document	Location
SNOMED CT File Naming Convention - 090623	Collaborative site
SNOMED CT Release Format 2 – Data Structures Specification	Collaborative site

Review Timetable

Review date	Responsible owner	Comments
20120631	CTO	Review after 24 month period

Table of Contents

1 Introduction

1.1 Purpose

This document describes the reference set specifications released as part of the SNOMED CT Release Format 2. This format is not mandated for internal terminology development usage or as an interchange mechanism between terminology development systems.

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 format specification was developed by harmonizing proposals reviewed by International Health Terminology Standards Development Organisation (IHTSDO) Enhanced Release Format Project Group, including:

- Enhanced Release Format Specification (International Health Terminology Standards Development Organisation. SNOMED Clinical Terms® Enhanced Release Format Proposed Specification, 21 June 2007)
- Reference Set Specification (International Health Terminology Standards Development Organisation. SNOMED Clinical Terms® Reference Sets - Proposed Specification, 31 July 2007)
- Alternate Release Format proposed by the National E-Health Transition Authority (NEHTA) in coordination with their Australian Affiliates

Please note that a glossary is not provided within this document. Instead, the glossary on the IHTSDO website should be referenced when that becomes available.

1.2 Who should read this specification?

The intended audience for this document 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 section.

This document should be read in conjunction with the "SNOMED CT Release Format 2 – Data Structures Specification" document, which provides a specification for the underlying data structures required to support RF2 and reference sets.

1.3 Associated Quality Measures

The definition of quality measures to monitor the implementation of this standard do not fall under the scope of this document, but will be published as part of the documentation covering the QA and Release process for the IHTSDO Workbench.

1.4 Separation of Reference Sets into Release files

Separation of reference sets into files may be done in a number of ways. Each reference set has a particular structure for the optional fields that are appended to each member. For example, a simple reference set has no additional fields; a CSI reference set has three additional fields - the first a Component, the second a String, and the third an Integer. There must be at least one reference set member file for each different reference set structure, as defined above. Reference sets may be further split, if required, by the owner of the reference sets. The naming conventions for the reference set files provide more detail on the naming convention to be used in this case (see the "SNOMED CT File Naming Convention" document).

Each reference set file has a header row containing field names for each of the columns. The names of the standard fields are the field names as detailed in the "SNOMED CT Release Format 2 – Data Structures Specification" document.

The additional fields are named "Attribute 1", "Attribute 2", "Attribute 3", etc. Alternatively, where only one Reference Set is released in a file, the additional fields may also be given names matching those provided in the Reference Set's Descriptor.

2 Reference Set Specifications

2.1 Overview

2.1.1 Introduction

This section first details how reference sets themselves are described in a machine readable form, using a set of |Reference set descriptor| member records (called a Descriptor, for short). It then describes a number of standard reference set patterns. Each of these patterns is also described in a machine readable form using a set of |Reference set descriptor| member records (called a Descriptor Template, for short). Each pattern may be used to define a number of reference sets. At the end of the section, a number of individual reference sets are described that do not conform to a particular pattern.

In each subsection, each reference set or reference set pattern is described in turn:

- The purpose of each reference set is first described;
- The format of the reference set member record is detailed in a table;
- The metadata supporting the reference set is described;
- The machine readable reference set descriptor member records for the reference set pattern (the Descriptor Template, for short) are then shown;
- Examples of usage are given, providing example Descriptors, where appropriate.

The first reference set to be described is the reference set descriptor. Subsequent sections describe a number of reference set patterns.

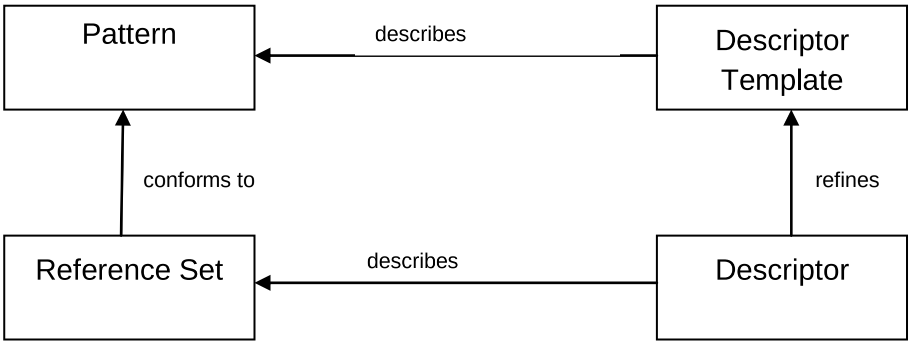
2.1.2 Descriptors, Descriptor Templates and Patterns

The purpose of the |Reference set descriptor| is to describe the format of all other reference sets that may be included in a release. A Descriptor held within the |Reference set Descriptor| describes the referencedComponentId field and the additional fields for the reference sets it describes. Each field is described using a concept in the metadata. The type of each field is also described in the same way.

Patterns allow a number of different types of reference set to be defined, each of which conforms to the specified pattern, having the same release file format. The file format of each reference set pattern is described by a Descriptor Template. This Descriptor Template describes the format and number of additional fields held against members of reference sets conforming to the pattern, and provides an envelope within which those additional fields may be further refined for each reference set conforming to the pattern. The Descriptor Template for each pattern is provided in the section describing that pattern.

Each defined reference set that conforms to a pattern has its own Descriptor, that describes its own specific properties, and although reference set field types must still conform to the Descriptor Template for the pattern, each field type may be further constrained using data sub-types specified in the metadata hierarchy. This provides some level of refinement to the constraints that may be applied to a reference set conforming to a particular pattern.

The following diagram shows the relationships between patterns, reference sets, Descriptor Templates and Descriptors graphically:



2.1.3 Patterns and Use Cases

The next table summarises the use cases for reference sets (one per row) that are described in the following sections, and shows which reference set patterns are used in each case:

		Patterns							
Use cases	|Attribute value type| (C)	|Simple map type| (S)	|Complex map type| (IISSSC)	|Language type| (C)	|Query Specification type|(CCS)	|Annotation type| (S)	|Association type| (C)	|Ordered type| (IC)	|Simple type|
Refinability of relationships	*								
ICD-10 mapping	*		*						
Inactivation indicator	*								
CVT3 map		*							
SNOMEDID map		*							

Language dialect				*					
Language dialect with context				*					
Intension reference set specification					*				
Image annotation						*			
Short annotation						*			
Descriptive annotation						*			
Reason for inactivation							*		
RF1 Subset representation				*				*	

Note: the letters shown after each pattern indicate the type and number of additional fields held against each member of a reference set conforming to that pattern, where 'C' is short for Component, 'S' is short for String and 'I' is short for Integer. For example, reference sets conforming to the |Attribute value type| (C) pattern has one additional field held against each member, a component reference; reference sets conforming to the |Simple type| pattern have no additional fields held against each member.

2.1.4 Metadata supporting Reference Sets

Reference sets are described by concepts under the |Reference set| sub-hierarchy. Values that can be used within reference set fields are described in the |Reference set attribute| sub-hierarchy. Both the |Reference set| sub-hierarchy and the |Reference set attribute| sub-hierarchy are held within the |Foundation metadata concept| sub-hierarchy which is part of the metadata hierarchy shown below. How each of the concepts shown in this metadata hierarchy is used is described in each of the following sections:

- SNOMED CT Model Component
  - Foundation metadata concept
    - Reference set attribute
      - Referenced component
      - Map correlation value
        - Not mappable
        - Exact match
        - Narrow SNOMED CT to broad target
        - Broad SNOMED CT to narrow target
        - Partial overlap
        - Not specified
      - Map target
      - Annotation
        - Image
      - Association source component
      - Attribute type
        - Component type
          - Concept type component
          - Description type component
          - Reference set member type component
          - Relationship type component
        - Integer
          - Signed integer

    - Unsigned integer
  - String
    - Time
    - URL
      - HTML reference
      - Image reference
        - JPEG reference
        - GIF reference
    - Text
      - Single character
      - Text < 256 bytes
    - UUID
- Acceptability
  - Acceptable
  - Preferred
- Map advice
- Map group
- Annotated component
- Association target component
- Attribute description
- Attribute order
- Attribute value
  - Order
  - Linked to
  - Refinability value
    - Not refinable
    - Optional refinability
    - Mandatory refinability
  - Inactive value
    - Limited
    - Duplicate
    - Ambiguous
    - Moved elsewhere
    - Outdated
    - Inappropriate
    - Erroneous
  - Active value
    - Pending move
    - Concept non-current
  - Description inactivation value
    - Limited
    - Duplicate
    - Pending move
    - Moved elsewhere
    - Concept non-current
    - Outdated
    - Inappropriate
    - Erroneous
  - Concept inactivation value
    - Limited
    - Duplicate
    - Pending move
    - Ambiguous
    - Moved elsewhere
    - Outdated
    - Erroneous
- Description length

  - Generated reference set
  - Target effective time
  - Description in dialect
  - Description format
    - DITA
    - Limited HTML
    - Plain text
    - XHTML
  - Source effective time
  - Query
  - Scheme value
  - Map priority
  - Map rule
  - Map source concept
- Reference set
  - Simple type
  - Ordered type
  - Association type
    - Historical association
      - POSSIBLY EQUIVALENT TO association reference set
      - REFERS TO concept association reference set
      - SIMILAR TO association reference set
      - MOVED FROM association reference set
      - MOVED TO association reference set
      - ALTERNATIVE association reference set
      - WAS A association reference set
      - REPLACED BY association reference set
      - SAME AS association reference set
  - Reference set descriptor
  - Language type
    - English
      - GB English
      - US English
  - Annotation type
    - Associated image
  - Description format
  - Module dependency
  - Query specification type
    - Simple query specification
  - Attribute value type
    - Relationship inactivation indicator reference set
    - Description inactivation indicator reference set
    - Relationship refinability reference set
    - Concept inactivation indicator reference set
  - Simple map type
    - SNOMED RT ID simple map
    - CTV3 simple map

2.1.5 Naming conventions for Reference Sets

National Release Centres and others may create additional reference sets. A namespace is required to create a new reference set, as each reference set is defined by a Concept. The Concept’s Fully Specified Name (FSN) and a Synonym are used to name the reference set. Where a new reference set is created against an existing pattern, then the following naming convention should be used (where the text "My particular" should be replaced by the name of the reference set):

Attribute value type reference set (pattern)
FSN = My particular attribute value reference set (foundation metadata concept)
PT = My particular reference set

Simple Map type reference set (pattern)
FSN = My particular simple map reference set (foundation metadata concept)
PT = My particular simple map

Complex Map type reference set (pattern)
FSN = My particular complex map reference set (foundation metadata concept)
PT = My particular complex map

Language type reference set (pattern) – for a Language refset
FSN = English - ISO 639-1 code 'en' language reference set (foundation metadata concept)
PT = English

Language type reference set (pattern) – for a Dialect RefSet
FSN = GB English language reference set (foundation metadata concept)
PT = GB English

Query specification type reference set (pattern)
FSN = My particular query specification reference set (foundation metadata concept)
PT = My particular query specification reference set

Annotation type reference set (pattern)
FSN = My particular annotation reference set (foundation metadata concept)
PT = My particular annotation reference set

Association type reference set (pattern)
FSN = My particular association reference set (foundation metadata concept)
PT = My particular association reference set

2.2 Reference set descriptor

2.2.1 Purpose

This reference set is used to describe the format of all other reference sets that are included in a release. The data type and meaning of the referenced component and each additional field within each reference set is described by this reference set.

Reference set descriptor can be used to define:

- The order of appearance of additional attributes (other than those mandatory for a reference set)
- The name and purpose of the additional attributes
- The data types for the additional attributes

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

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

2.2.2 Reference Set Data Structure

This CCI (Component-Component-Integer) reference set has the following format:

Field	Data type	Purpose
id	UUID	
effectiveTime	Time	
active	Boolean	
moduleId	SctId	
refSetId	SctId	Set to the |Reference set descriptor| concept in the metadata hierarchy.
referencedComponentId	SctId	To set a descendant of |Reference set| in the metadata hierarchy.
attributeDescription	SctId	To set a descendant of |Reference set attribute| in the metadata hierarchy, that describes the additional attribute extending the reference set.

Field	Data type	Purpose
attributeType	SctId	To set a descendant of |Attribute type| in the metadata hierarchy, that describes the type of the additional attribute extending the reference set.
attributeOrder	Integer	An unsigned integer, providing an ordering for the additional attributes extending the reference set. A zero value here is used for the row that describes the referenced component within the reference set. One additional active row exists for each attribute extending the reference set.

At least one row must exist for each reference set included in a release. This row must have an attributeOrder value of '0' and an attributeType of 'Component type' (or one of its descendents). The referenceComponentId then describes the member (or referenced component) of the reference set.

One additional row exists to describe each additional attribute that extends the reference set under consideration.

Creation of a refset descriptor is mandatory when authoring a refset in the International release or an NRC extension.

Otherwise, creation of a refset descriptor is optional. Where a refset descriptor is not defined for a reference set, then the closest ancestor of the reference set that has a refset descriptor can be used when validating reference set member records.

2.2.3 Supporting Metadata

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

- Foundation metadata concept
  - Reference set attribute
    - Attribute type
      - Component type
        - Concept type component
        - Description type component
        - Reference set member type component
        - Relationship type component
      - Integer
        - Signed integer
        - Unsigned integer
      - String
        - Time

- URL
  - HTML reference
  - Image reference
    - JPEG reference
    - GIF reference
- Text
  - Single character
  - Text < 256 bytes
- UUID
- Attribute description
- Attribute order
- Attribute value
- Reference set
  - Reference set descriptor

2.2.4 Descriptor

The following table shows the |Reference set descriptor| active member entries for the |Reference set descriptor| itself:

refSetId	referencedComponentId	attributeDescription	attributeType	attributeOrder
|Reference set descriptor|	|Reference set descriptor|	|Reference set|	|Concept type component|	0
|Reference set descriptor|	|Reference set descriptor|	|Reference set attribute|	|Concept type component|	1
|Reference set descriptor|	|Reference set descriptor|	|Attribute type|	|Concept type component|	2
|Reference set descriptor|	|Reference set descriptor|	|Attribute order|	|Unsigned integer|	3

In the above and following examples, descriptions have been used in place of component identifiers for clarity.

Within a particular release, attributeOrders are contiguous for a particular referencedComponentId within a reference set descriptor.

2.3 Simple type reference set (pattern)

2.3.1 Purpose

This reference set pattern allows a collection of components to be grouped together.

2.3.2 Reference Set Data Structure

A reference set, without any optional fields is be used to support the Simple pattern.

Field	Data type	Purpose
id	UUID	A 128 bit unsigned integer, uniquely identifying the reference set member.
effectiveTime	Time	Specifies the inclusive date at which this change becomes effective.
active	Boolean	Specifies whether the member's state was active or inactive from the nominal release date specified by the effectiveTime field.
moduleId	SctId	Identifies the member version's module. Set to a child of |Module| within the metadata hierarchy.
refSetId	SctId	Set to a child of |Simple type| in the metadata hierarchy.
referencedComponentId	SctId	A reference to the SNOMED CT component to be included in the reference set.

2.3.3 Supporting Metadata

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

```
SNOMED CT Model Component
        Foundation metadata concept
                Reference set
                        Simple type
```

2.3.4 Descriptor Template

One reference set descriptor member is required for each instance of the Simple type Reference Set.

The table below holds the Descriptor Template for the |Simple type| reference set pattern:

refSetId	referencedComponentId	attributeDescription	attributeType	attributeOrder
|Reference set descriptor|	|Simple type|	|Referenced component|	|Component type|	0

2.4 Ordered type reference set (pattern)

2.4.1 Purpose

This reference set pattern allows a collection of components to be defined, and optionally given a priority ordering, and split into a number of sub-groups, if necessary.

It can be used to represent Navigation, Duplicate terms, Realm concept, Realm relationship and Context concept reference sets.

2.4.2 Reference Set Data Structure

An Integer Component reference set is used to support the Ordered type pattern.

Field	Data type	Purpose
id	UUID	A 128 bit unsigned integer, uniquely identifying the reference set member.
effectiveTime	Time	Specifies the inclusive date at which this change becomes effective.
active	Boolean	Specifies whether the member's state was active or inactive from the nominal release date specified by the effectiveTime field.
moduleId	SctId	Identifies the member version's module. Set to a child of |Module| within the metadata hierarchy.
refSetId	SctId	Set to a child of |Ordered type| in the metadata hierarchy.
referencedComponentId	SctId	A reference to the SNOMED CT component being tagged with a value.
order	Integer	The priority order of this item in the set, where a value of '1' is the highest priority, and higher values are of lower priority. A value of '0' is not allowed.

Field	Data type	Purpose
linkedTo	SctId	This field either enables members of the reference set to be linked into a number of sub-groups, or enables a number of children concepts to be linked to a single parent concept. To link members into a sub-group, all components in the same sub-group should reference the component in the group that has an order of '1' (i.e. – the highest priority component). Therefore, all components that have the same linkedTo value are in the same sub-group. To link a number of children concepts to a single parent concept, one member record should exist per child, with the referencedComponentId field referencing the parent and this field referencing the child concept. The order field is then used to order the children concepts under the parent concept. For members that are not linked in either of the above ways, this field should be set to '0'.

2.4.3 Supporting Metadata

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

```
SNOMED CT Model Component
    Foundation metadata concept
        Reference set attribute
            Order
            Linked to
        Reference set
            Ordered type
```

2.4.4 Descriptor Template

One group of reference set descriptor members are required for each instance of the |Ordered type| Reference Set.

The table below holds the Descriptor Template for the |Ordered type| reference set pattern:

refSetId	referencedComponentId	attributeDescription	attributeType	attributeOrder
|Reference set descriptor|	|Ordered type|	|Referenced component|	|Component type|	0
|Reference set descriptor|	|Ordered type|	|Order|	|Unsigned integer|	1
|Reference set descriptor|	|Ordered type|	|Linked to|	|Component type|	2

2.5 Attribute value type reference set (pattern)

2.5.1 Purpose

This reference set pattern allows a value from a specified range to be associated with a component.

2.5.2 Reference Set Data Structure

A Component reference set is used to support the attribute value pattern.

Field	Data type	Purpose
id	UUID	A 128 bit unsigned integer, uniquely identifying the reference set member.
effectiveTime	Time	Specifies the inclusive date at which this change becomes effective.
active	Boolean	Specifies whether the member's state was active or inactive from the nominal release date specified by the effectiveTime field.
moduleId	SctId	Identifies the member version's module. Set to a child of |Module| within the metadata hierarchy.
refSetId	SctId	Set to a child of |Attribute value type| in the metadata hierarchy.
referencedComponentId	SctId	A reference to the SNOMED CT component being tagged with a value.
valueId	SctId	Set to a grandchild of |Attribute value|.

2.5.3 Supporting Metadata

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

```
SNOMED CT Model Component
        Foundation metadata concept
                Reference set attribute
                        Referenced component
                        Attribute value
                                ICD-10 map category value
```

      - Map source concept is outside of the scope of target classification
      - Map source concept is properly classified
      - Map source concept cannot be classified with available data
      - Map of source concept is context dependent
      - Source SNOMED concept is ambiguous
      - Source SNOMED concept is incompletely modeled
      - Mapping guidance from WHO is ambiguous
      - Source concept has been retired from map scope
    - Refinability value
      - Not refinable
      - Optional refinability
      - Mandatory refinability
    - Inactive value
      - Limited
      - Duplicate
      - Ambiguous
      - Moved elsewhere
      - Outdated
      - Inappropriate
      - Erroneous
    - Active value
      - Pending move
      - Concept non-current
    - Description inactivation value
      - Limited
      - Duplicate
      - Pending move
      - Moved elsewhere
      - Concept non-current
      - Outdated
      - Inappropriate
      - Erroneous
    - Concept inactivation value
      - Limited
      - Duplicate
      - Pending move
      - Ambiguous
      - Moved elsewhere
      - Outdated
      - Erroneous
- Reference set
  - Attribute value type
    - Relationship inactivation indicator reference set
    - Description inactivation indicator reference set
    - Relationship refinability reference set
    - Concept inactivation indicator reference set
    - ICD-10 map category reference set

No status indicators for “Current” and “Retired without reason” have been included, as reference set members are not used in these cases.

2.5.4 Descriptor Template and example Descriptors

One group of reference set descriptor members is required for each type of attribute value reference set.

The table below holds the Descriptor Template for the |Attribute value type| reference set pattern:

refSetId	referencedComponentId	attributeDescription	attributeType	attributeOrder
|Reference set descriptor|	|Attribute value type|	|Referenced component|	|Component type|	0
|Reference set descriptor|	|Attribute value type|	|Attribute value|	|Concept type component|	1

The table below holds the Descriptor for the |ICD-10 map category reference set|:

refSetId	referencedComponentId	attributeDescription	attributeType	attributeOrder
|Reference set descriptor|	|ICD-10 map category reference set|	|Referenced component|	|Reference set member type component|	0
|Reference set descriptor|	|ICD-10 map category reference set|	|ICD-10 map category value|	|Concept type component|	1

This table holds the Descriptor for the Relationship refinability reference set|:

refSetId	referencedComponentId	attributeDescription	attributeType	attributeOrder
|Reference set descriptor|	|Relationship refinability reference set|	|Referenced component|	|Relationship type component|	0
|Reference set descriptor|	|Relationship refinability reference set|	|Refinability value|	|Concept type component|	1

This table holds the Descriptor for the |Concept inactivation indicator reference set|:

refSetId	referencedComponentId	attributeDescription	attributeType	attributeOrder
|Reference set descriptor|	|Concept inactivation indicator reference set|	|Referenced component|	|Concept Type component|	0
|Reference set descriptor|	|Concept inactivation indicator reference set|	|Concept inactivation value|	|Concept Type component|	1

This table holds the Descriptor for the |Description inactivation indicator reference set|:

refSetId	referencedComponentId	attributeDescription	attributeType	attributeOrder
|Reference set descriptor|	|Description inactivation indicator reference set|	|Referenced component|	|Description type component|	0
|Reference set descriptor|	|Description inactivation indicator reference set|	|Description inactivation value|	|Concept type component|	1

Similar members also exist for the relationship inactivation indicator and the reference set inactivation indicator.

2.5.5 Example Usage – Refinability of Relationships

This example shows how a relationship's refinability is modelled using reference set member records:

refSetId	referencedComponentId	valueId
|Relationship refinability reference set|	*A SNOMED CT Relationship*	|Not refinable|
|Relationship refinability reference set|	*A SNOMED CT Relationship*	|Optional refinability|
|Relationship refinability reference set|	*A SNOMED CT Relationship*	|Mandatory refinability|

A relationship (referenced by the referencedComponentId field) can be associated with an enumeration concept (a child of the |Refinability value| concept in the metadata hierarchy), held in the valueId field.

2.5.6 Example Usage – ICD-10 map categories

This example shows attribute value members for an ICD-10 map category reference set:

refSetId	referencedComponentId	valueId
|ICD-10 map category reference set|	*ICD-10 map refset member Id*	|Source SNOMED concept is ambiguous|
|ICD-10 map category reference set|	*ICD-10 map refset member Id*	|Map source concept is properly classified|

A Reference Set member (referenced by the referencedComponentId field) can be associated with an enumeration concept (a child of the |ICD-10 map category value| concept in the metadata hierarchy), held in the valueId field.

Note that in the above example, *ICD-10 map refset member Id* would be the UUID that identifies the Reference Set member record.

2.5.7 Example Usage – Inactivation indicator

This example shows attribute value members for the inactivation indicator reference sets:

refSetId	referencedComponentId	valueId
|Concept inactivation indicator reference set|	*Concept 1*	|Duplicate|
|Description inactivation reference set|	*Description 1*	|Concept non-current|

For the purposes of the above example, assume that Description 1 is an active description linked to Concept 1, which is inactive (this is not shown in the table). The reference set members indicate that the reason for Concept 1's inactivation was because it was a duplicate concept and Description 1 is a valid description on the inactive concept.

2.6 Simple map type reference set (pattern)

2.6.1 Purpose

This reference set pattern supports simple maps between SNOMED CT concepts and values in alternate coding schemes. No constrains are put on the number of coding schemes supported, the number of codes within a particular scheme mapped to by a single SNOMED CT concept or the number of SNOMED CT concepts mapping to a particular code. However, this reference set should primarily be used when there is a reasonably close "one-to-one" mapping between SNOMED CT concepts and the alternate coding scheme.

2.6.2 Reference set data structure

A String reference set is used to support simple maps.

Field	Data type	Purpose
id	UUID	A 128 bit unsigned integer, uniquely identifying the reference set member.
effectiveTime	Time	Specifies the inclusive date at which this change becomes effective.
Active	Boolean	Specifies whether the member's state was active or inactive from the nominal release date specified by the effectiveTime field.
moduleId	SctId	Identifies the member version's module. Set to a child of |Module| within the metadata hierarchy.
refSetId	SctId	Set to a child of the |Simple map type| in the metadata hierarchy. This identifies the alternate scheme that is being mapped to.
referencedComponentId	SctId	A reference to the SNOMED CT concept being mapped.
mapTarget	String	The value of the code in the alternate scheme being mapped to.

2.6.3 Metadata

The following metadata hierarchy supports this reference set:

- SNOMED CT Model Component
    - Foundation metadata concept
        - Referenced component
        - Scheme value
        - Reference set
            - Simple map type
                - SNOMED RT ID simple map
                - CTV3 simple map

2.6.4 Descriptor Template and example Descriptors

One group of reference set descriptor members is required for each type of simple map reference set.

The table below holds the Descriptor Template for the simple map reference set pattern:

refSetId	referencedComponentId	attributeDescription	attributeType	attributeOrder
|Reference set descriptor|	|Simple map type|	|Referenced component|	|Concept type component|	0
|Reference set descriptor|	|Simple map type|	|Scheme Value|	|String|	1

This table holds the "reference set descriptor" active member entries for a |CTV3 simple map| reference set:

refSetId	referencedComponentId	attributeDescription	attributeType	attributeOrder
|Reference set descriptor|	|CTV3 simple map|	|Referenced component|	|Concept type component|	0
|Reference set descriptor|	|CTV3 simple map|	|Scheme value|	|String|	1

In the above and following examples, descriptions have been used in place of component identifiers for clarity.

2.6.5 Example usage

This table holds example entries for two reference sets conforming to this pattern:

refSetId	referencedComponentId	mapTarget
|CTV3 simple map|	10006000	72710
|CTV3 simple map|	100060003	XU014
|SNOMED RT ID simple map|	10006000	P1-A68A3

|SNOMED RT ID simple map|	100060003	C-D1777 1

2.7 Complex map type reference set (pattern)

2.7.1 Purpose

This reference set pattern supports more complex maps from each SNOMED CT concept to one or more codes in a target scheme, where each target code may be selected at run-time from a number of alternate codes, based on either a set of machine readable rules or human readable advice.

2.7.2 Reference Set Data Structure

An IISSSC (Integer-Integer-String-String-String-Component) reference set is used to support complex maps:

Field	Data type	Purpose
id	UUID	A 128 bit unsigned integer, uniquely identifying the reference set member.
effectiveTime	Time	Specifies the inclusive date at which this change becomes effective.
active	Boolean	Specifies whether the member's state was active or inactive from the nominal release date specified by the effectiveTime field.
moduleId	SctId	Identifies the member version's module. Set to a child of |Module| within the metadata hierarchy.
refSetId	SctId	Set to one of the children of the |Complex map type| concept in the metadata hierarchy.
referencedComponentId	SctId	A reference to the SNOMED CT concept being mapped.
mapGroup	Integer	An integer, grouping a set of complex map records from which one may be selected as a target code. Where a SNOMED CT concept maps onto 'n' target codes, there are 'n' groups, each containing one or more complex map records.

Field	Data type	Purpose
mapPriority	Integer	Within a group, the mapPriority specifies the order in which complex map records should be checked. Only the first map record meeting the run-time selection criteria is taken as the target code within the group of alternate codes.
mapRule	String	A machine-readable rule, (evaluating to either 'true' or 'false' at run-time) that indicates whether this map record should be selected within its mapGroup
mapAdvice	String	Human-readable advice, that may be employed by the software vendor to give an end-user advice on selection of the appropriate target code from the alternatives presented to him within the group.
mapTarget	String	The target code in the scheme to be mapped onto.
correlationId	SctId	A child of |Map correlation value| in the metadata hierarchy, identifying the correlation between the SNOMED CT concept and the target code.

Values for the mapGroup field are allocated on a sequential basis (for each refSetId and referencedComponentId combination) starting from '1', but are not necessarily sequential, as groups may be created and removed during a mapping process that may straddle several releases. For maps where each SNOMED CT concept only maps to at most one of a group of alternate target codes, the mapGroup field should be set to '1'.

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

The mapRule and mapAdvice fields enable run-time selection (within vendor's software) from a number of alternative map records within a mapGroup. Where alternatives are not required, these fields should be set to null. Where complex maps are required, either, both or neither of these fields may be populated.

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

2.7.3 Supporting Metadata

The following metadata supports this reference set:

- SNOMED CT Model Component
  - Foundation metadata concept
    - Reference set attribute
      - Map correlation value
        - Not mappable
        - Exact match
        - Narrow SNOMED CT to broad target
        - Broad SNOMED CT to narrow target
        - Partial overlap
        - Not specified
      - Map target
      - Map advice
      - Map group
      - Map priority
      - Map rule
      - Map source concept
    - Reference set
      - Complex map type
        - ICD-10 complex map

2.7.4 Descriptor Template

The table below holds the Descriptor Template for this reference set pattern:

refSetId	referencedComponentId	attributeDescription	attributeType	attributeOrder
|Reference set descriptor|	|Complex map type|	|Map source concept|	|Concept type component|	0
|Reference set descriptor|	|Complex map type|	|Map group|	|Unsigned integer|	1
|Reference set descriptor|	|Complex map type|	|Map priority|	|Unsigned integer|	2
|Reference set descriptor|	|Complex map type|	|Map rule|	|String|	3
|Reference set descriptor|	|Complex map type|	|Map advice|	|String|	4
|Reference set descriptor|	|Complex map type|	|Map target|	|String|	5
|Reference set descriptor|	|Complex map type|	|Map correlation value|	|Concept type component|	6

In the above and following examples, descriptions have been used in place of component identifiers for clarity.

2.7.5 Example Usage

The following example (using simplified ids and codes for clarity) shows how a |Complex map type| reference set can be used in conjunction with an |Attribute value type| reference set to produce an ICD-10 map:

id	refSetId	referenced ComponentId	map Group	map Priority	mapAdvice	mapTarget	Correlation Id
11	|ICD-10 complex map|	A	5	1	If X true, then	P	|Not specified|
12	|ICD-10 complex map|	A	5	2	Otherwise	Q	|Not specified|
13	|ICD-10 complex map|	A	6	1	If Y true, then	R	|Not specified|
14	|ICD-10 complex map|	A	6	2	Otherwise	S	|Not specified|

refSetId	Referenced ComponentId	valueId
|ICD-10 map category|	11	|Map source concept is properly classified|
|ICD-10 map category|	12	|Map source concept is properly classified|
|ICD-10 map category|	13	|Map source concept is properly classified|
|ICD-10 map category|	14	|Map source concept is properly classified|

The table holds four ICD-10 map records (with UUIDs 11-14), mapping from SNOMED CT concept A to two ICD-10 target codes (represented by the two groups 5 and 6).

Human readable map advice indicates that the SNOMED CT concept A should map to both:

- ICD-10 code "P", if X is true; but otherwise to ICD-10 code "Q"; and to
- ICD-10 code "R", if Y is true; but otherwise to ICD-10 code "S".

The associated attribute value reference set indicates that the source concept has been properly classified.

2.7.6 Rule Language specification

A specification of a grammar for the rule language is required. This grammar should:

- Evaluate to true, false or indeterminate (if there is a lack of data).
- Be sensitive to the context in which it is being used in the vendor's system. The grammar should include access to a number of variables that can be bound (at run-time, within the vendor's system) to appropriate attributes in the information model.

The definition of this rule grammar is outside the scope of this specification.

2.8 Language type reference set (pattern)

2.8.1 Purpose

This reference set pattern supports the creation of sets of descriptions for one or more dialects of a language, perhaps for use within a particular context.

A general mechanism to enable members of one reference set to be based on members of another reference set will be provided in a future addition to this specification.

2.8.2 Reference set data structure

A Component reference set is used to support language reference sets.

Field	Data type	Purpose
id	UUID	A 128 bit unsigned integer, uniquely identifying the reference set member.
effectiveTime	Time	Specifies the inclusive date at which this change becomes effective.
active	Boolean	Specifies whether the member's state was active or inactive from the nominal release date specified by the effectiveTime field.
moduleId	SctId	Identifies the member version's module. Set to a child of |Module| within the metadata hierarchy.
refSetId	SctId	A descendant of |Language type| in the metadata hierarchy
referencedComponentId	SctId	A reference to the Description included in the language reference set.
acceptabilityId	SctId	A descendant of |Acceptability| in the metadata hierarchy.

Within each language reference set, there must be at most one Description (for each included concept) with a typeId of |Fully specified name|. Additionally, there must be one and only one Description (for each included concept) with a typeId of |Synonym| that has an acceptabilityId field (within the reference set) of |Preferred|.

2.8.3 Metadata

The following metadata supports this reference set:

- SNOMED CT Model Component
  - Foundation metadata concept
    - Reference set attribute
      - Description in dialect
      - Acceptability
        - Acceptable
        - Preferred
    - Reference set
      - Language type
        - English
          - GB English
            - Pathology (GB English)
          - US English

In the above |Language type| sub-hierarchy, each reference set may represent a particular dialect or language relevant to a particular context.

The immediate children of |Language type| are languages. This level may be used to represent the "correct" language, where a language authority exists. In most cases, however, this level is likely to be empty.

2.8.4 Descriptor Template

One group of reference set descriptor members are required for each language reference set. The table below holds the Descriptor Template for the language reference set pattern:

refSetId	referencedComponentId	attributeDescription	attributeType	attributeOrder
|Reference set descriptor|	|Language type|	|Description in dialect|	|Description type component|	0
|Reference set descriptor|	|Language type|	|Acceptability|	|Concept type component|	1

The table below holds the Descriptor for the "GB English" reference set:

refSetId	referencedComponentId	attributeDescription	attributeType	attributeOrder
|Reference set descriptor|	|GB English|	|Description in dialect|	|Description type component|	0
|Reference set descriptor|	|GB English|	|Acceptability|	|Concept type component|	1

2.8.5 Example usage

This table holds example entries for this reference set:

refSetId	referencedComponentId	acceptabilityId	active
|GB English|	Autopsied body	|Preferred|	1
|GB English|	Autopsied body, NOS	|Acceptable|	1

2.9 Query specification type reference set (pattern)

2.9.1 Purpose

This reference set pattern allows a serialised query grammar to be associated with a reference set to enable the generation of its members. The specified query is run over the entire SNOMED CT hierarchy to produce the resultant reference set.

The resultant reference set need not be a simple reference set. Given a query language that is expressive enough, any number of optional fields may be associated with the resultant member records.

2.9.2 Reference Set Data Structure

A String reference set may be used to support serialised query specifications in the RF2 release format.

Field	Data type	Purpose
id	UUID	A 128 bit unsigned integer, uniquely identifying the reference set member.
effectiveTime	Time	Specifies the inclusive date at which this change becomes effective.
active	Boolean	Specifies whether the member's state was active or inactive from the nominal release date specified by the effectiveTime field.
moduleId	SctId	Identifies the member version's module. Set to a child of |Module| within the metadata hierarchy.
refSetId	SctId	A child of |Query specification type| in the metadata hierarchy.
referencedComponentId	SctId	The reference set for which members are to be generated.
query	String	The serialised query that can be used to (re-)generate the reference set members.

2.9.3 Supporting Metadata

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

```
SNOMED CT Model Component
    Foundation metadata concept
        Reference set attribute
            Generated reference set
            Query
        Reference set
            Query specification type
                Simple query specification
                CS query specification
```

2.9.4 Descriptor Template

One group of reference set descriptor members are required for each type of query. The table below holds the Descriptor Template for the |Query specification type| reference set pattern:

refSetId	referencedComponentId	attributeDescription	attributeType	attributeOrder
|Reference set descriptor|	|Query specification type|	|Generated reference set|	|Concept type component|	0
|Reference set descriptor|	|Query specification type|	|Query|	|String|	1

The table below holds the Descriptor for the |CS query Specification| reference set:

refSetId	referencedComponentId	attributeDescription	attributeType	attributeOrder
|Reference set descriptor|	|CS query Specification|	Generated reference set	|Concept type component|	0
|Reference set descriptor|	|CS query Specification|	Query	|String|	1

2.9.5 Example Usage

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

refSetId	referencedComponentId	query
|Simple query specification|	*Reference set 1*	“serialised query 1”
|CS query specification|	*Reference set 2*	“serialised query 2”

“serialised query 2”, however, must also generate a Component id and a String value for each reference set member that it generates for *Reference set 2*. In this case, *Reference set 2* is a CS reference set.

2.9.6 Query language specification

The specification of the query language has yet to be defined / selected, but it should be capable of:

- Selecting concepts using primary fields, subsumption testing, relationships, relationship groups, set operators (union, intersection, excludes), and lexical query.
- Selecting descriptions, relationships and reference sets using similar mechanisms.
- Calculation of values for the reference set’s extended fields, identifying the version of the syntax and any language syntax variations.
- Expressing ref set query definitions for terminologies other than SNOMED CT. The syntax should not assume that the only target is SNOMED CT, it should allow at least for ICDx, LOINC, ICPC, and local vocabularies, particularly lab related.

The definition of the query language is outside the scope of this specification.

2.10 Annotation type reference set (pattern)

2.10.1 Purpose

The annotation reference set pattern allows strings to be associated with components for a particular purpose.

2.10.2 Reference Set Data Structure

A String reference set is used to support annotations.

Field	Data type	Purpose
id	UUID	A 128 bit unsigned integer, uniquely identifying the reference set member.
effectiveTime	Time	Specifies the inclusive date at which this change becomes effective.
active	Boolean	Specifies whether the member's state was active or inactive from the nominal release date specified by the effectiveTime field.
moduleId	SctId	Identifies the member version's module. Set to a child of |Module| within the metadata hierarchy.
refSetId	SctId	A child of |Annotation type| in the metadata hierarchy.
referencedComponentId	SctId	A reference to the component to be annotated
annotation	String	The annotation to attach to the component.

2.10.3 Supporting Metadata

The following metadata in supports this reference set:

- SNOMED CT Model Component
  - Foundation metadata concept
    - Reference set attribute
      - Annotated component
      - Annotation
        - Image
    - Reference set
      - Annotation type

Associated image

2.10.4 Descriptor Template

One group of reference set descriptor members are required for each annotation reference set.

The table below holds the Descriptor Template for the annotation reference set pattern:

refSetId	referencedComponentId	attributeDescription	attributeType	attributeOrder
|Reference set descriptor|	|Annotation type|	|Annotated component|	|Concept type component|	0
|Reference set descriptor|	|Annotation type|	|Annotation|	|String|	1

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

The table below holds the Descriptor for the |Associated image| annotation reference set, which allows URLs to be associated with concepts:

refSetId	referencedComponentId	attributeDescription	attributeType	attributeOrder
|Reference set descriptor|	|Associated image|	|Annotated component|	|Concept type component|	0
|Reference set descriptor|	|Associated image|	|Image|	|URL|	1

Note that in the table above, the |URL| concept is a descendent of |String| concept in the metadata.

2.10.5 Example Usage

This table holds example entries for this reference set:

refSetId	referencedComponentId	Annotation
|Associated image|	*Concept 1*	"http://example.com/picture.jpeg"
|Associated image|	*Concept 2*	"http://example.com/picture.gif"

In the above example, the two URLs have been used to annotate two SNOMED CT concepts.

It is not recommended that this mechanism be used to annotate concepts with text that may require translation to other languages. Instead, such text should be included under an appropriate description type within the Description file.

2.11 Association type reference set (pattern)

2.11.1 Purpose

This reference set pattern allows associations of particular types to be described between two components.

2.11.2 Reference Set Data Structure

A Component reference set is used to support associations.

Field	Data type	Purpose
id	UUID	A 128 bit unsigned integer, uniquely identifying the reference set member.
effectiveTime	Time	Specifies the inclusive date at which this change becomes effective.
active	Boolean	Specifies whether the member's state was active or inactive from the nominal release date specified by the effectiveTime field.
moduleId	SctId	Identifies the member version's module. Set to a child of |Module| within the metadata hierarchy.
refSetId	SctId	A descendant of |Association type| in the metadata hierarchy.
referencedComponentId	SctId	A reference to the source component of the association.
targetComponentId	SctId	A reference to the destination component of the association.

2.11.3 Supporting Metadata

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

```
SNOMED CT Model Component
        Foundation metadata concept
                Reference set attribute
                        Association source component
```

    - Association target component
- Reference set
    - Association type
        - Historical association
            - POSSIBLY EQUIVALENT TO association reference set
            - REFERS TO concept association reference set
            - SIMILAR TO association reference set
            - MOVED FROM association reference set
            - MOVED TO association reference set
            - ALTERNATIVE association reference set
            - WAS A association reference set
            - REPLACED BY association reference set
            - SAME AS association reference set

2.11.4 Notes on usage

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

Each |Historical association| reference set holds relationships of a different nature between the Components. The |Historical association| reference sets contains associations:

- 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 reference set for which there is a current replacement to the replacement reference 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 each inactive concept to one or more concepts that replace it.

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

Within the |REFERS TO concept association reference set|, the referencedComponentId field must be a Description and the targetComponentId must be a Concept.

The targetComponentId is used differently in the |MOVED TO association reference set|. In this case, the targetComponentId does not refer directly to a replacement component, but rather to the namespace to which the component was moved to. The targetComponentId actually refers to the concept that represents the namespace. 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.

2.11.5 Descriptor Template and Descriptor examples

One group of reference set descriptor members are required for each association type reference set.

The table below holds the Descriptor Template for the |Association type| reference set pattern:

refSetId	referencedComponentId	attributeDescription	attributeType	attributeOrder
|Reference set descriptor|	|Association type|	|Association source component|	|Component type|	0
|Reference set descriptor|	|Association type|	|Association target component|	|Component type|	1

The table below holds the Descriptor for the |POSSIBLY EQUIVALENT TO association reference set||. Each member of this reference set identifies a target Concept that may describe the source concept, which is ambiguous:

refSetId	referencedComponentId	attributeDescription	attributeType	attributeOrder
|Reference set descriptor|	POSSIBLY EQUIVALENT TO association reference set	|Association source component|	|Concept component type|	0
|Reference set descriptor|	POSSIBLY EQUIVALENT TO association reference set	|Association target component|	|Concept component type|	1

The table below holds the Descriptor for the |REFERS TO concept association reference set|. Each member of this reference set identifies a target Concept that is correctly described by the Term of the inactive source Description:

refSetId	referencedComponentId	attributeDescription	attributeType	attributeOrder
|Reference set descriptor|	|REFERS TO concept association reference set|	|Association source component|	|Description type component|	0
|Reference set descriptor|	|REFERS TO concept association reference set|	|Association target component|	|Concept type component|	1

The table below holds the Descriptor for the |SIMILAR TO association reference set|. Each member of this reference set identifies a target Description that is identical in all respects except for the associated Term which, while not identical, is similar to the source Description:

refSetId	referencedComponentId	attributeDescription	attributeType	attributeOrder
|Reference set descriptor|	|SIMILAR TO association reference set|	|Association source component|	|Description component type|	0
|Reference set descriptor|	|SIMILAR TO association reference set|	|Association target component|	|Description component	1

			type|	

The table below holds the Descriptor for the |MOVED FROM association reference set|. Members of this reference set identify the original target component in another namespace that is the origin of the source Component:

refSetId	referencedComponentId	attributeDescription	attributeType	attributeOrder
|Reference set descriptor|	|MOVED FROM association reference set|	|Association source component|	|Component type|	0
|Reference set descriptor|	|MOVED FROM association reference set|	|Association target component|	|Component type|	1

The table below holds the Descriptor for the |MOVED TO association reference set|. Members of this reference set identify the target Concept representing the Namespace to which the source Component has been moved:

refSetId	referencedComponentId	attributeDescription	attributeType	attributeOrder
|Reference set descriptor|	|MOVED TO association reference set|	|Association source component|	|Component type|	0
|Reference set descriptor|	|MOVED TO association reference set|	|Association target component|	|Concept component type|	1

The table below holds the Descriptor for the |ALTERNATIVE association reference set|. Members of this reference set may identify one of several target components that are alternatives that are similar or equivalent to the source Component (e.g. where a single Component is replaced by two more narrowly defined Components):

refSetId	referencedComponentId	attributeDescription	attributeType	attributeOrder
|Reference set descriptor|	|ALTERNATIVE association reference set|	|Association source component|	|Component type|	0
|Reference set descriptor|	|ALTERNATIVE association reference set|	|Association target component|	|Component type|	1

The table below holds the Descriptor for the |WAS A association reference set|. Members of this reference set identify a target concept that is a revised replacement for the source concept:

refSetId	referencedComponentId	attributeDescription	attributeType	attributeOrder
|Reference set descriptor|	|WAS A association reference set|	|Association source component|	|Concept component type|	0
|Reference set descriptor|	|WAS A association reference set|	|Association target component|	|Concept component type|	1

The table below holds the Descriptor for the |REPLACED BY association reference set|. Members of this reference set identify a target component that is a revised replacement for the source component:

refSetId	referencedComponentId	attributeDescription	attributeType	attributeOrder
|Reference set descriptor|	|REPLACED BY association reference set|	|Association source component|	|Component type|	0
|Reference set descriptor|	|REPLACED BY association reference set|	|Association target component|	|Component type|	1

The table below holds the Descriptor for the |SAME AS association reference set|. Members of this reference set identify a target component that is an identical duplicate of the source component:

refSetId	referencedComponentId	attributeDescription	attributeType	attributeOrder
|Reference set descriptor|	|SAME AS association reference set|	|Association source component|	|Component type|	0
|Reference set descriptor|	|SAME AS association reference set|	|Association target component|	|Component type|	1

2.11.6 Example Usage – Replaced by

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

refSetId	referencedComponentId	targetComponentId
|REPLACED BY association reference set|	*Concept 1*	*Concept 2*
|REPLACED BY association reference set|	*Concept 3*	*Concept 4*

In this example, the associations describe that Concept 1 has been replaced by Concept 2 and Concept 3 has been replaced by Concept 4.

2.11.7 Example Usage – Refers to Concept

The following table holds example entries for the |REFERS TO concept association reference set| reference set.

refSetId	referencedComponentId	targetComponentId
|REFERS TO concept association reference set|	Desc1	Concept 3

|REFERS TO concept association reference set|	Desc2	Concept 4

In this example, the associations identify that Concept 3 is correctly described by the Term of the inactive Description, Desc1 and Concept 4 is correctly described by the Term of the inactive Description, Desc2.

2.12 Module Dependency Reference Set

2.12.1 Purpose

This reference set allows dependencies between module versions to be described.

2.12.2 Reference Set Data Structure

A String-String reference set is used to support module dependencies.

Field	Data type	Purpose
id	UUID	A 128 bit unsigned integer, uniquely identifying the reference set member.
effectiveTime	Time	Specifies the inclusive date at which this change becomes effective.
active	Boolean	Specifies whether the member's state was active or inactive from the nominal release date specified by the effectiveTime field.
moduleId	SctId	Identifies the member version's module. Set to a child of |Module| within the metadata hierarchy.
refSetId	SctId	A reference to the |Module dependency| concept in the metadata hierarchy.
referencedComponentId	SctId	A reference to the module that this module is dependent on, a descendant of |Module| in the metadata hierarchy.
sourceEffectiveTime	String	The effective time of the source module. This allows a specific module version to be selected as having a dependency. The effectiveTime must match exactly.
targetEffectiveTime	String	The effective time of the target module. This allows a specific module version to be selected as being the subject of a dependency. The effectiveTime must match exactly.

2.12.3 Supporting Metadata

The following metadata supports this reference set:

- SNOMED CT Model Component
  - Core metadata concept
    - Module
      - IHTSDO maintained module
        - SNOMED CT core
        - SNOMED CT model component
  - Foundation metadata concept
    - Reference set attribute
      - Target effective time
      - Source effective time
    - Reference set
      - Module dependency

Each component within a SNOMED CT release references a moduleId. This is the module that the component is currently mastered in (from the effectiveTime held on the component record). A module is simply a collection of SNOMED CT components that are maintained as a unit by a single organisation. It is the organisation's responsibility to organise the components in each extension that it is responsible for into one or more modules, in a way that best fits its business needs.

A module is modelled by a descendent of the |Module| concept in the metadata hierarchy. The |Module| sub-hierarchy is organised by a maintaining organisation into a number of groups. For example, all modules maintained by IHTSDO are children of |IHTSDO maintained module|. The |Module| sub-hierarchy models modules maintained by each organisation and does NOT model module dependencies. Instead, module dependencies are modelled using the |Module dependency| reference set.

At the point of release, if any component within a module has changed, then a new row is added for the module's concept, with the effectiveTime set to the date of the new release, irrespective of whether the other fields in the module concept record itself have changed. The updated |Module| concept record identifies that some components within the module have been updated in this release. Where no components within a module have been updated, then a new module record is not added and the module's effectiveTime field does not change from the previous release.

Each SNOMED CT component is in one, and only one module. The module that a component is mastered in may change over time, and when this happens, the component's moduleId field is updated (in the usual way by appending a row for the component).

Each module is in one and only one extension. Modules do not straddle extensions. The extension that a module resides in is defined by the SctId of the module. A module may not move from one extension to another over time. If the components within a module are to be moved to another extension, then a new module must be created within the destination extension to host the components that are to be transferred.

There may be more than one module in an extension.

2.12.4 Descriptor

The table below holds the “reference set descriptor” active member entries for the |Module dependency| reference set:

refSetId	referencedComponentId	attributeDescription	attributeType	attributeOrder
|Reference set descriptor|	|Module dependency|	|Module|	|Concept type component|	0
|Reference set descriptor|	|Module dependency|	|Source effective time|	|Time|	1
|Reference set descriptor|	|Module dependency|	|Target effective time|	|Time|	2

2.12.5 Example Usage

Module version dependencies are modelled using a reference set. A module version may depend on one or more other module versions, and many module versions may have a dependency on a single module version. Cyclic module version dependencies are not allowed. The table below holds example entries for the module dependencies reference set:

refSetId	moduleId	Referenced ComponentId	Source EffectiveTime	Destination EffectiveTime
|Module dependency|	|SNOMED CT Australian extension|	|SNOMED CT core|	T2	T1
|Module dependency|	|SNOMED CT Australian Pathology|	|SNOMED CT Australian extension|	T2	T2
|Module dependency|	|SNOMED CT Australian Pathology|	|SNOMED CT core|	T2	T1
|Module dependency|	|SNOMED CT Australian Discharge summary|	|SNOMED CT Australian extension|	T2	T2
|Module dependency|	|SNOMED CT Australian Discharge summary|	|SNOMED CT core|	T2	T1

All dependencies are described in the release files, not just immediate dependencies. It is the responsibility of the organisation owning a dependent module to identify all modules on which it depends. Therefore, the |Module dependency| reference set members are held within the dependent module. This is why the moduleId of the reference set member record is always the source module.

In the above example, the dependencies describe that the |SNOMED CT Australian Pathology| and the |SNOMED CT Australian Discharge Summary| module versions released at T2 are both dependent on the |SNOMED CT Australian extension| module version in the same release, which is itself dependent on the |SNOMED CT core| module version released at T1.

Any release should consist of a set of module versions that are certified as being compatible. Each release should also identify other existing module versions that are outside the scope of the release, but that the release is dependent on.

As dependencies between module versions are described (not just dependencies between modules), it is possible to describe a dependency from a current module in a release to a version of a module in a previous release, if so desired. It is also possible to correct historical dependencies between previous modules if these had previously been stated incorrectly.

2.13 Description Type Reference Set

2.13.1 Purpose

This reference set provides format and maximum length information for each description type.

2.13.2 Reference Set Data Structure

The CI (Component-Integer) reference set format is described below:

Field	Data type	Purpose
id	UUID	A 128 bit unsigned integer, uniquely identifying the reference set member.
effectiveTime	Time	Specifies the inclusive date at which this change becomes effective.
active	Boolean	Specifies whether the member's state was active or inactive from the nominal release date specified by the effectiveTime field.
moduleId	SctId	Identifies the member version's module. Set to a child of |Module| within the metadata hierarchy.
refSetId	SctId	Set to the |Description format| reference set concept in the metadata hierarchy.
referencedComponentId	SctId	A reference to a child of |Description type| in the metadata hierarchy
descriptionFormat	SctId	A reference to a child of |Description format| reference set attribute concept in the metadata hierarchy.
descriptionLength	Integer	The maximum length in bytes for descriptions of this description type.

2.13.3 Supporting Metadata

The following metadata in the |Foundation metadata concept| hierarchy supports the description format reference set:

- SNOMED CT Model Component
  - Core metadata concept
    - Description type
      - Definition
      - Fully specified name
      - Synonym
      - Purpose
  - Foundation metadata concept
    - Reference set attribute
      - Description format
        - DITA
        - Limited HTML
        - Plain text
        - XHTML
      - Description length
    - Reference set
      - Description format

2.13.4 Descriptor

The table below holds the Descriptor for the Description Format reference set:

refSetId	referencedComponentId	attributeDescription	attributeType	attributeOrder
|Reference set descriptor|	|Description format|	|Description type|	|Concept type component|	0
|Reference set descriptor|	|Description format|	|Description format|	|Concept type component|	1
|Reference set descriptor|	|Description format|	|Description length|	|Unsigned integer|	2

2.13.5 Example Usage

This example holds the entries for the Description Format reference set:

refSetId	referencedComponentId	descriptionFormat	descriptionLength
|Description format|	|Fully specified name|	|Limited HTML|	1024
|Description format|	|Synonym|	|Limited HTML|	1024
|Description format|	|Definition|	|Limited HTML|	1024
|Description format|	|Purpose|	|Limited HTML|	1024

3 Implementation Advice

This section describes the steps required to perform a number of common activities.

3.1 How to create a new Reference Set using an existing pattern

In order to create a new Reference Set, you need access to a namespace in order to generate SctIds. Within your namespace, you should add one moduleId concept (with an FSN and a Synonym), under the |Module| sub-hierarchy within the metadata, for each of your authoring organisations.

Then, follow the steps below to create a new reference set:

3.1.1 Define the Reference Set in the metadata hierarchy

First, create a concept for the Reference Set:

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 organisation.
definitionStatusId	SctId	|Primitive|

Then, add up to three Descriptions for the FSN, the Preferred Term and optionally the Purpose:

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

Field	Data type	Set to
		authoring organisation.
conceptId	SctId	The identifier of the concept describing the Reference Set that you've just added.
languageCode	String	The language of the Description.
typeId	SctId	Create up to three descriptions, with each of the following types: |FSN|, |Synonym|, |Purpose|. The first two are mandatory, the third is optional.
term	String	Terms for the FSN, the Synonym and the Purpose. The Synonym is the string used to commonly refer to the Reference Set. The conventions for creating terms for the FSN and Synonym terms are described in Section 2.
caseSignificanceId	SctId	|Case Sensitive|

Add an "Is a" Relationship to link the Reference Set to the appropriate pattern:

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 organisation.
sourceId	SctId	The identifier of the concept describing the Reference Set that you've just added.
destinationId	SctId	The concept describing the pattern that this Reference Set follows, a descendant of |Reference Set| in the metadata hierarchy.
relationshipGroup	Integer	'0'
typeId	SctId	|Is a|

Field	Data type	Set to
characteristicTypeId	SctId	|Stated relationship|
modifierId	SctId	|Some|

3.1.2 Define the Reference Set Attributes within the metadata hierarchy

Add new concepts for each of the Reference Set member attributes, if necessary. If the Reference Set attributes describing the pattern are adequate to describe the Reference Set's attributes, then these can be used instead, and you can skip to the next section.

You may wish to create your own Reference Set attributes for one of the following reasons:

- You wish to give one or more of the attributes a different name than that of the pattern.
- You wish to make the purpose of a particular attribute more explicit in the metadata.
- You wish to limit the set of allowed values for one or more of the attributes.
- You wish to make the type of one or more of the attributes more specific than that given in the pattern.

You may add new concepts for some of the attributes, and reuse existing concepts for other attributes, if you wish.

For each attribute that you wish to create, first add a concept:

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 organisation.
definitionStatusId	SctId	|Primitive|

Then, link it with an |Is a| Relationship into the |Reference set attribute| metadata hierarchy.

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'

Field	Data type	Set to
moduleId	SctId	The module identifier for your authoring organisation.
sourceId	SctId	The identifier of the concept describing the Reference set attribute that you've just added.
destinationId	SctId	|Reference set attribute|
relationshipGroup	Integer	'0'
typeId	SctId	|Is a|
characteristicTypeId	SctId	|Stated relationship|
modifierId	SctId	|Some|

Then, add up to three Descriptions (for FSN, Preferred Term and optionally Purpose) for each of the new attributes:

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 organisation.
conceptId	SctId	The identifier of the concept describing the attribute that you've just added.
languageCode	String	The language of the Description.
typeId	SctId	Create up to three Descriptions for each new attribute, with the following types: |FSN|, |Synonym|, |Purpose|. The first two are mandatory, the third is optional.
term	String	Terms for the FSN, a Synonym and the Purpose. The Synonym is the string used to commonly refer to the attribute (and therefore should appear as a column header in tables showing the Reference Set member records).
caseSignificanceId	SctId	|Case Sensitive|

If any of the Reference Set member attributes are to be limited to a range of values, then add a concept for each allowed value in the range, and link the concept using an |Is a| relationship to the member attribute. Then add two Descriptions for the FSN and Preferred Term of each allowed attribute value.

In order to limit the range of an attribute, it must have a type of |Concept type component| (as held in the attributeType field of the Descriptor – see the next section).

For each allowed value that an attribute can take, add a concept:

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 organisation.
definitionStatusId	SctId	|Primitive|

Then, link it with an |Is a| Relationship into the attribute that you've just added in the |Reference set attribute| metadata hierarchy.

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 organisation.
sourceId	SctId	The identifier of the concept describing the allowed attribute value that you've just added.
destinationId	SctId	The identifier of the concept describing the attribute that you've just added.
relationshipGroup	Integer	'0'
typeId	SctId	|Is a|
characteristicTypeId	SctId	|Stated relationship|
modifierId	SctId	|Some|

And finally, add two Descriptions for the allowed attribute value concept:

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 organisation.
conceptId	SctId	The identifier of the concept describing the allowed attribute value that you've just added.
languageCode	String	The language of the Description.
typeId	SctId	Create two descriptions, with each of the following types: |FSN|, |Synonym|
term	String	Terms for the FSN and a Synonym. The Synonym is the string used to commonly refer to the allowed attribute value (and therefore should be the one shown in pick lists used when maintaining Reference Set member records).
caseSignificanceId	SctId	|Case Sensitive|

3.1.3 Create the Descriptor for the Reference Set

Add one record to the |Reference Set Descriptor| Reference Set describing the referencedComponentId attribute, and one additional row for each additional optional attribute within the Reference Set.

These records together describe the structure of the Reference Set, and are called the Descriptor of the reference set, for short. If the existing Descriptor Template (that describes the Reference Set's pattern) also adequately describes the reference set that you've just created, then a new Descriptor need not be created, and this section may be skipped.

Where a Descriptor is created for a new Reference Set, it should have the same structure (i.e. – an identical number of records, each of the same attribute type or subtype) as the Reference Set Descriptor that described the parent Reference Set pattern.

Field	Data type	Purpose
id	UUID	A unique UUID for this record.
effectiveTime	Time	The nominal date of release for your reference set.

Field	Data type	Purpose
active	Boolean	'1'
moduleId	SctId	The module identifier for your authoring organisation.
refSetId	SctId	|Reference set descriptor|
referencedComponentId	SctId	Set to the concept describing the Reference Set that you've just created.
attributeDescription	SctId	Set to the concept describing the attribute that you've just created, or alternatively an existing concept under the |Reference set attribute| metadata hierarchy.
attributeType	SctId	To set a descendant of |Attribute type| in the metadata hierarchy. This field describes the type of the attribute. If an attribute has been limited to a range of values, then this field must always be set to |Concept type component|. Where a Reference Set is the child of a Reference Set pattern, then this field must be the same as or a descendant of the equivalent field for the Reference Set pattern.
attributeOrder	Integer	Add one record with the attributeOrder field set to '0' (describing the referencedComponentId attribute column of the Reference Set). Then add one additional record for each optional attribute in the Reference Set (with values of '1', '2', '3', etc.).

3.1.4 Add members to the Reference Set

Follow the steps in the next section to maintain the members of the Reference set.

3.2 How to add, change or remove members of an existing Reference Set

In order to add, remove or modify members in a Reference Set, follow the guidance below.

To add a member to an existing Reference Set, create a new record as follows:

Field	Data type	Purpose

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 organisation.
refSetId	SctId	The id of the concept that describes the Reference Set that you’re adding a member to.
referencedComponentId	SctId	A reference to a component, of type (and possibly range) limited by the Descriptor record for this Reference Set with attributeOrder ‘0’.
additional field 1		An optional attribute, with a value, of type (and possibly range) limited by the Descriptor record for this Reference Set with attributeOrder ‘1’.
additional field 2		An optional attribute with a value, of type (and possibly range) limited by the Descriptor record for this Reference Set with attributeOrder ‘2’.
...		

To delete an existing member from a Reference Set, create a new record as follows:

Field	Data type	Purpose
id	UUID	A unique UUID of the existing member record that you wish to delete.
effectiveTime	Time	The nominal date of release in which this member is to be deleted.
active	Boolean	‘0’
moduleId	SctId	The module identifier for your authoring organisation.
refSetId	SctId	As value in existing record
referencedComponentId	SctId	As value in existing record
additional field 1		As value in existing record

Field	Data type	Purpose
additional field 2		As value in existing record
…		

To modify an existing member in a Reference Set, create a new record as follows:

Field	Data type	Purpose
Id	UUID	A unique UUID for the existing member record that is to be updated.
effectiveTime	Time	The nominal date of release that the update is to become active in.
active	Boolean	'1'
moduleId	SctId	The module identifier for your authoring organisation.
refSetId	SctId	As value in existing record. A member cannot move from one reference set to another.
referencedComponentId	SctId	As value in existing record. A member cannot change the component that it refers to. Instead, the existing member record should be deleted, and a new one created.
additional field 1		This field may be updated. An optional attribute, with a value, of type (and possibly range) limited by the Descriptor record for this Reference Set with attributeOrder '1'.
additional field 2		This field may be updated. An optional attribute with a value, of type (and possibly range) limited by the Descriptor record for this Reference Set with attributeOrder '2'.
…		

Note: reference set members should only be amended under the authority of the owner of the reference set.

3.3 How to create a new Reference Set pattern

In order to create a new reference set pattern, follow the steps to create a new reference set, with the following exceptions:

- The concept describing the Reference Set pattern should be created as an immediate child of the |Reference set| concept, or as a child of another Reference Set pattern.
- The Descriptions of typeId |Synonym| and |FSN| should be of the form:
    - *My pattern name* type
    - *My pattern name* type reference set (foundation metadata concept)
- A Descriptor Template must be created for a pattern, following the steps as described to create a Descriptor for a Reference Set.

3.4 How to use a reference set

3.4.1 Consuming a Reference Set

One or more Reference Sets may be held in a single Reference Set release file. However, if there are more than one Reference Sets in a single file, they all have the same structure (i.e. – the same number of additional fields of the same top level types of Component, Integer or String).

Each record in the Reference Set file represents a member of the reference set. The refSetId column identifies the Reference Set that the member record belongs to.

The refSetId is an SctId that can be used to look up the concept in the |Reference Set| metadata that describes the reference set. Up to three Descriptions (with three different typeIds) may be associated with the Reference Set concept:

- A Description with a typeId of |FSN|, used to formally describe the Reference Set. This Description always exists.
- A Description with a typeId of |Synonym|, used to name the Reference Set. This Description always exists, and can be used to display the name of the Reference Set within a system.
- A Description with a typeId of |Purpose|, used to describe the purpose of the Reference Set. This Description may or may not be present.

The refSetId can also be used to look up the Reference Set Descriptor, in the |Reference set descriptor| Reference Set. This can be done by identifying the member records in the |Reference set descriptor| reference set with a referencedComponentId that matches the refSetId of the Reference Set.

There is one Descriptor record describing the referencedComponentId field in the Reference Set and one additional record for each optional field within the Reference Set. The Descriptor record with an attributeOrder field value of ‘0’ describes the referencedComponentId field; a Descriptor record with an attributeOrder field value of ‘1’ would describe the first optional field; etc.

For each Reference set field being described (i.e. - the referencedComponentId and each optional field), two fields in the Descriptor record provide additional information:

- The attributeType field is a reference to a concept under the |Attribute type| metadata hierarchy that provides typing information for the field. At the top level, this could be |Component type|, |Integer| or |String|, and would then match the typing information available within the Reference Set file name (see the file naming convention document). However, the type of a field can also be specified at a finer level of granularity using the attributeType field. For instance, instead of the attributeType being specified simply as an |Integer|, it may instead be specified as an |Unsigned integer| or a |Signed integer|. For a full list of types, see the |Attribute type| metadata hierarchy.
- The attributeDescription field is a reference to a concept under the |Reference set attribute| metadata hierarchy that also provides additional information about each Reference Set field. Up to three Descriptions (with three different typeIds) may be associated with each of these concepts:
    - A Description with a typeId of |FSN|, used to formally describe the Reference Set field. This Description always exists.
    - A Description with a typeId of |Synonym|, used to name the Reference Set field. This Description always exists, and can be used to display a column header for each Reference Set field used within a system.
    - A Description with a typeId of |Purpose|, used to describe the purpose of the Reference Set field. This Description may or may not be present.

Additionally, if the attributeType is |Concept type component|, then the children of the concept referred to by the attributeDescription provide a list of allowed concept enumeration values for the Reference Set field. Each of these concepts has two Descriptions with typeIds of |FSN| and of |Synonym|, and the latter set of Descriptions can be used to validate field entry for concept enumeration type Reference Set fields or to create pick-lists to allow users to select one or more values. Where the attributeDescription concept does not have any children, then no limitation is placed on the concepts allowed in the Reference Set field.

3.4.2 Using Reference Sets without Descriptors

All Reference Sets that are released from IHTSDO or from a National Release Centre have an associated Descriptor for the Reference Set. However, Descriptors are optional for other organisations that create Reference Sets. Where you are using a Reference Set for which a Descriptor has not been created, and you need additional information about the Reference Set, the Descriptor of the closest ancestor of the concept describing the Reference Set that does have a Descriptor may be used. This situation should be rare, as an organisation that releases Reference Sets should only release them without Descriptors if it is sure that its consumers do not require the information held within the Descriptors.

3.4.3 Using Reference Sets to hold simple value sets

Where you know that a single simple Reference Set is held in a file, then a simple value set may be retrieved from the Reference Set by taking the referencedComponentIds of each record with an active field set to ‘1’. Each value in the value set is then an SctId of a SNOMED CT component.

Where a release file contains multiple simple Reference Sets, then a number of value sets may be retrieved from the file by taking the referencedComponentIds of each record with an active field set to ‘1’, and grouping them into value sets by using the refSetId field. Each value in the value set is an SctId of a SNOMED CT component. In order to retrieve the name of each value set, its refSetId can be used to identify a |Reference set| metadata concept that has a Description with a typeId of |Synonym| that provides a name for the value set.