INTERNATIONAL HEALTH TERMINOLOGY
STANDARDS DEVELOPMENT ORGANISATION

# SNOMED CT Release Format 2.0 Guide for Updating from RF1 to RF2

Date 20100706
Version 1.0

## Amendment History

| Version | Date | Editor | Comments |
|---|---|---|---|
| 0.1 | 20090630 | John Gutai (IHTSDO) | Draft for comment, based on document produced from Salt Lake City meeting |
| 0.2 | 20090709 | John Gutai (IHTSDO) | Updated following review |
| 0.3 | 20090724 | John Gutai | Included comments from Community of Practice Review |
| 0.4 | 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 | Update 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 – Reference Set Specifications | 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 key changes in the SNOMED CT Release Format 2.0 (RF2) that was developed to resolve a number of known issues with the previous release format (RF1). A set of agreed principles (see Section 2) were applied to guide the development of the RF2 specification.

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 International Health Terminology Standards Development Organisation (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 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 guide?

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 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.

## 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 Summary of Changes

The RF2 introduces a number of new concepts and capabilities. These are summarised below, and described in more detail later in this document:

- 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 centres alike, the RF2 design is expected to result in long term savings as well as improvement in product functionality and quality.

## 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 has 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.

# 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 artefacts deemed in scope of the release format.
- Consistent identification of all components throughout their lifecycle and clear identification of all other artefacts 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 artefacts.
- 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 core SNOMED CT.
- 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 organisational 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.

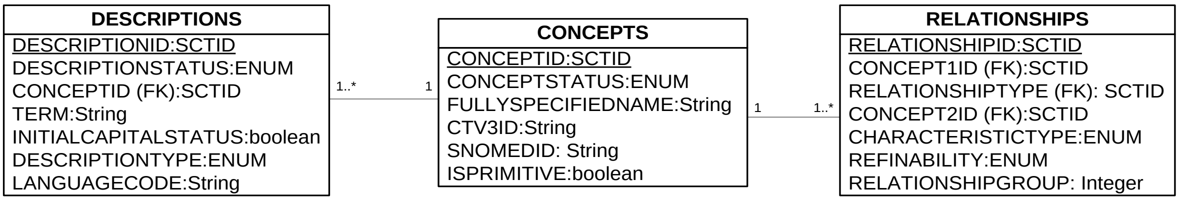
# 3 Rationale for moving from RF1 to RF2

## 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 on the current release format, refer to the SNOMED CT *Technical Reference Guide* and SNOMED CT *Technical Implementation Guide*.

The current release format used for the International SNOMED CT release is depicted in the following diagram.



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 cross maps to other terminologies and coding systems, history tracking of components and descriptions, subsets of SNOMED CT concepts, and other uses.

## 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 terminology itself.
- No consistent and clearly defined mechanism for release centres, developers, implementers, and end users to extend the core 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 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.

## 3.4 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 are be made available in both formats for a limited number of consecutive release cycles, for convenience. It is expected that National Release Centres are follow the same approach, and also release in dual format for a number of release cycles, unless there are specific reasons not to.

# 4 Details of Key Changes

The following subsections discuss details of the key changes between RF2 and the previous release format.

## 4.1 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 is supplied in the RF2 release files, dating back to the first release in the current format in 2002.

Once released, a row in any of the RF2 release files remains 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 effectiveTime field 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.

## 4.2 Active field

As mentioned above, each file contains a Boolean active field, used to indicate whether, after the point in time specified in the effectiveTime field, 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.

## 4.3 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) 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 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.

| 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)<br>Moved from ('6' in References table) | \|MOVED FROM association reference set\| |
| MOVED TO (in Relationships table)<br>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<br>Replaced by ('1' in References table) | \|REPLACED BY association reference set\| |
| SAME AS (in Relationships table)<br>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.

## 4.4 Field naming

Lower camel case has been used for field names in distribution file headers and in documentation that describes these files. File names 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.

## 4.5 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 effectiveTime gives the nominal release date at which this version of the component came into effect.
- The active flag states whether the component is 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.

## 4.6 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 uses concepts in a metadata hierarchy to represent the enumerated value set rather than using 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 standardised 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:

| **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:

| 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\| |
| | 0 | \|Stated relationship\| |
| | 1 | \|Qualifying relationship\| |
| | 2 | (no specified value) – now modelled through the inactive association reference set. |

# 4.7 Reference Set Data Structures

## 4.7.1 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.

### 4.7.2 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, taken from the Technical Reference Guide (TRG):

| 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.

| Field | Description |
|---|---|
| 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. |
| 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. |

Some Subsets and their members are generated automatically from an XML definition file.

### 4.7.3 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:

| **SubsetType value** | **Description** | **Mapping to RF2** |
|---|---|---|
| 1 | Language | Language type Reference Set |
| 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 |

### 4.7.4 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 ContextId fields 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:

| Field | Data type | Set to |
|---|---|---|
| id | SctId | A unique id in your namespace. |
| 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 effectiveTime of 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 organisation. |
| definitionStatusId | SctId | \|Primitive\| |

Then, add up two Descriptions for the FSN and the Preferred Term of the 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 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: \|FSN\|, \|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)". |
| caseSignificanceId | SctId | \|Case Sensitive\| |

Finally, add one Reference Set member record for each record in the Subset Members table for the Subset:

| 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 organisation. |
| 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. |
| linkedTo | SctId | Set to LinkedId 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:

| 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.

### 4.7.5 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 RealmId field 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 RealmId field in the Subset record).

The Reference Set member records should be created as described in the following table:

| 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 organisation. |
| refSetId | SctId | A reference to the concept describing the Reference Set that you've just created. |

| Field | Data type | How to populate |
|---|---|---|
| referencedComponentId | SctId | Set to MemberId in the Subset Members table record. |
| acceptabilityId | SctId | Set to \|Preferred\| if the MemberStatus field of the Subset Member record is either '1' (Preferred Term) or '3' (Fully specified name); set to \|Acceptable\| if the MemberStatus field of the Subset Member record is '2' (Synonym).<br>Note: all Context Description subset members should be therefore set up with an acceptabilityId of \|Preferred\|. |

A Descriptor can also be set up if required.

## 4.8 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 sub-hierarchy has been defined as a child of 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|) are moved to the |SNOMED CT Model Component| sub-hierarchy.

The |SNOMED CT Model Component| hierarchy is structured as follows:

- SNOMED CT Model Component
  - Core metadata concept
    (holds concept enumerations that are required for the core data structures)
  - Foundation metadata concept
    (holds metadata that is required by the extensibility mechanism)
  - Linkage
    (holds attribute types transferred from the linkages hierarchy)
  - Namespace
    (holds namespaces)

Note that only "Is a" relationships exist between concepts in the |SNOMED CT Model Component hierarchy|. Other associations between concepts in this hierarchy can be modelled using Association reference sets, described in more detail in the "SNOMED CT Release Format 2 – Reference Set Specifications" document.

## 4.9 SctIds and UUIDs

UUIDs are unique universal identifiers. These 128 bit unsigned integers can be used to identify all SNOMED CT components internally.

SctIds 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 reference concepts in clinical information systems and messages. SctIds are also used to identify relationships. However, UUIDs are used to identify Reference Set members.

In addition, any UUIDs used in development can also be published as additional identifiers via the Identifier file.

## 4.10 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 Description Type Reference Set allows a maximum length and format to be associated with each description type within the Description file.

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.

## 4.11 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.

## 4.12 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 be ***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 to 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 are 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 are need to be performed, and the results reviewed and approved.

The reason that this field has been included at this stage is that the RF2 format is likely to be stable for a long period of time, without addition or deletion of fields in the core tables. Although there is an extensibility mechanism, it was felt that, as this field is core to the semantics of the DL, it should be included as a core field from the start, and not added within the extensibility mechanism at a later date.

Any expansion of SNOMED CT to include relationships with a modifierId set to a value other than |Some| are be discussed with Members first and approved by the Technical Committee.

## 4.13 Addition of moduleId field

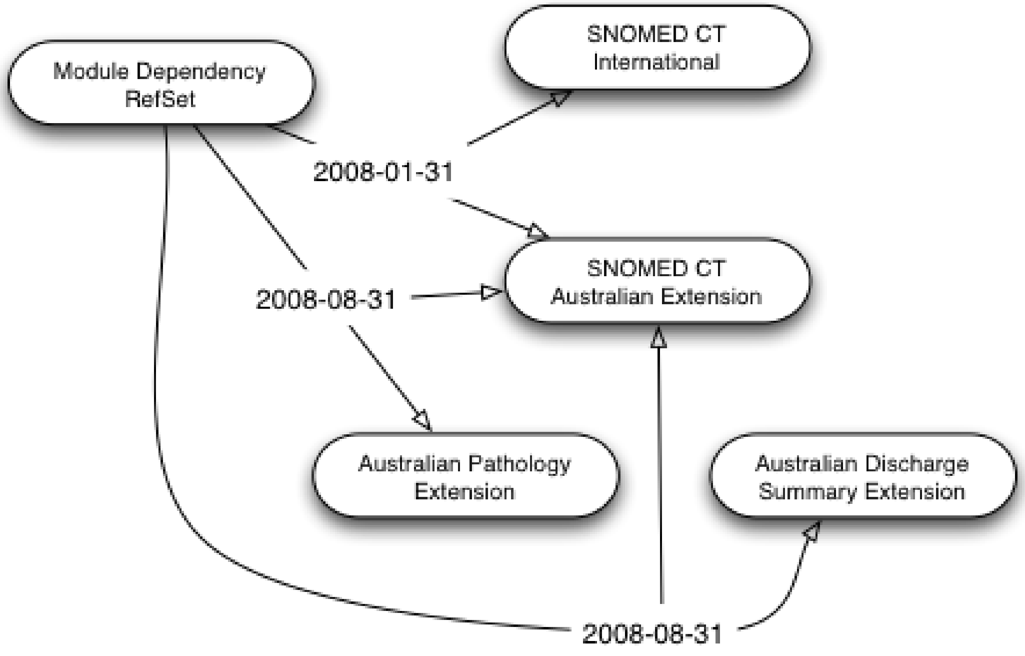
A moduleId field has been added to help identify content and dependencies in a release. This enables release centres to compose a unified release (in a single set of release files) from a number of different modules, yet still identify the origin of content down to a row level within each of the releases. 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 Set Specifications" 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


Module Dependency RefSet
SNOMED CT International
2008-01-31
SNOMED CT Australian Extension
2008-08-31
Australian Pathology Extension
Australian Discharge Summary Extension
2008-08-31

## 4.14 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 can 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.

## 4.15 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 is provided with the RF2 distribution.

| 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 ID simple 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\|. |

## 4.16 Identifier file

The Identifier file has been added to provide a standardised 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 organisational 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 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.

## 4.17 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 a number of Association reference sets (see the “History Tables” section in this document and the “SNOMED CT Release Format 2 – Reference Set Specifications” document for more detail).

## 4.18 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.

## 4.19 Mapping

### 4.19.1 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.

### 4.19.2 Representing RF1 cross maps in RF2

RF1 cross maps 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 cross map can be identified from the MapSetType field in the CrossMapSets table. The following values in the MapSetType field are possible:

| 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 one-to-one maps | None known of | Can be mapped automatically. Further definition are be given if necessary. |
| 4 | Alternate one-to-many | None known of | May need manual intervention to map. |

For cross maps 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 cross map to be represented in RF2 format. Where the MapSetRealmId field is blank or null, then an intermediate concept should not be created, and the cross map Reference Set concept should be created as a direct child of |Complex map|. The concept should be created as follows:

| Field | Data type | Set to |
|---|---|---|
| id | SctId | A unique id in your namespace. |
| effectiveTime | Time | The nominal date of release for your cross map reference set. The year of the nominal release should tie up with the year in the MapSetSchemeVersion field in the Cross Maps Sets record. |
| active | Boolean | '1' |
| moduleId | SctId | The module identifier for your authoring organisation. |
| definitionStatusId | SctId | \|Primitive\| |

Once the concept is created, add two Descriptions for the FSN and a Synonym.

| 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 Reference Set 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 the Synonym. The Synonym should be set to the MapSetName in the Cross Maps Sets record. The FSN should be set to: MapSetScemeName + " (" + MapSetSchemeId + ")" + " reference set (foundation metadata concept)". |
| caseSignificanceId | SctId | \|Case Sensitive\| |

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 cross map that you're representing in RF2. For each CrossMaps table record, identify the related CrossMapTarget record using the MapTargetId field in the CrossMaps record. The TargetCodes field in the CrossMapTarget record are 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:

| 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’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 |
| mapAdvice | String | Set to null |
| mapTarget | String | Set to the target code in the TargetCodes field of the CrossMapTargets record. |
| correlationId | StdId | For the ICD-9-CM map, set using the MapAdvice field in the CrossMaps record if this is not null; otherwise, set to the MapAdvice field in the CrossMapTargets record.<br>The following mapping should be used to set this field to the appropriate \|Map correlation value\| in the metadata hierarchy:<br>MapAdvice -> CorrelationId<br>‘0’ -> \|Not mappable\|<br>‘1’ -> \|Exact match\|<br>‘2’ -> \|Narrow SNOMED CT to broad target\|<br>‘3’ -> \|Broad SNOMED CT to narrow target\|<br>‘4’ -> \|Partial overlap\|<br>‘5’ -> \|Not specified\|<br>For other maps, set to \|Not specified\|. |

## 4.20 Release types

Given RF2's history tracking capability, it is possible to perform a number of different releases of content:

- A "full" release of each file, containing every version of every component ever released.
- A "snapshot" release, containing only the most recent version of every component ever released (whether this be active or inactive).
- A "delta" release, containing only component versions created since the last release. These component versions may represent a new component or a change to an existing component.

There are of course valid use cases for each type of release form. Following the first draft for trial use release, each International release are incorporate all three of these release types, allowing users to choose the most appropriate format for their needs.

A full release should always be available from release centres. Optionally, other release formats may also be made available. Where out of cycles releases are made, these follow the same format as standard cycle releases.

## 4.21 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 Editions 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 are require revisions to the SIF specification.

## 4.22 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 are 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 are also need to be extended to cater for this.