Leading healthcare terminology, worldwide

SNOMED CT Extensions Practical Guide

v1.0

Latest web browsable version: http://snomed.org/extpg

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

Publication date: 2018-07-03

Table of Contents

SNOMED International

Leading healthcare terminology, worldwide

The Extensions Practical Guide presents the purpose, process and principles of creating, distributing and managing SNOMED CT extensions. Extension producers should be familiar with this guidance to ensure the quality and integrity of the SNOMED CT editions that they publish.

This guide has been developed to assist National Release Centers and other extension producers to develop extensions based on recommended principles.

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

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

1 Executive Summary

SNOMED CT is a multilingual clinical terminology that covers a broad scope. However, some users may need additional concepts, relationships, descriptions or reference sets to support national, local or organizational needs. SNOMED CT is designed to allow the International Edition to be enhanced by creating extensions that meet national or local requirements. The extension mechanism allows SNOMED CT to be customized to address the terminology needs of a country or organization that are not met by the International Edition. An extension may contain components (i.e. concepts, descriptions or relationships) and/or reference sets used to represent subsets, maps or language preferences.

SNOMED CT extensions can support a variety of use cases, including:

- Translating SNOMED CT, for example
  - Adding terms used in a local language or dialect
  - Adding terms used by a specific user group, such as patient-friendly terms
- Configuring the terminology for specific use cases, for example
  - Specifying pick lists to be used for data entry
  - Specifying groups of components for reporting and analytics
  - Linking components to clinical knowledge resources
- Managing content gaps, for example
  - Adding components that are missing in the International Edition
  - Adding concepts that are only relevant to a local context
- Mapping between SNOMED CT and other code systems, for example
  - Representing maps between SNOMED CT concepts and codes from other code systems
- Extending the expressivity of SNOMED CT, for example
  - Extending the concept model by introducing new attributes to meet specific data retrieval use cases

The logical design of a SNOMED CT extension is technically consistent with that of the International Edition. Both represent and version SNOMED CT components and reference sets in release files that conform to the Release Format 2 specification. Every SNOMED CT extension includes one or more modules, and each module contains either SNOMED CT components or reference sets (or both). A SNOMED CT extension is published as a SNOMED CT edition, which includes the contents of a focus module from the extension, together with the contents of all the modules on which it depends. This includes the modules in the International Edition and possibly other modules from a national and/or local extension. National and local extensions are managed by SNOMED International Members or Affiliates who have been issued a namespace identifier by SNOMED International. A namespace identifier is used to create globally unique SNOMED CT identifiers for each component (i.e. concept, description and relationship) and reference set within a Member or Affiliate extension. This ensures that references to extension concepts contained in health record data are unambiguous and can be clearly attributed to a specific issuing organization. Once a namespace identifier has been obtained from SNOMED International, an extension producer must create one or more module concepts, which will be used to organize the extension content. All components and reference set members in an extension must belong to a module created by the responsible organization.

The key steps in producing a SNOMED CT extension are:

1. Planning - To ensure that the requirements can be met by the extension design, resources and processes that will be used.
2. Preparation - To ensure that the technical prerequisites are in place, including the namespace, modules, module dependencies and appropriate tooling.
3. Production - To develop, distribute and maintain the extension, including:
   a. Assessing requests - Requests for new terminology products or content changes are assessed to determine whether or not they will be accepted into the extension or submitted for national or international consideration.
   b. Authoring - SNOMED CT components and reference set members are added, modified or inactivated according to the SNOMED CT editorial principles and policies. It should be noted that:

- It is the responsibility of the extension producer to ensure that the quality and integrity of the extension is maintained, and that all content changes are made in a module that is owned by the *terminology producer* themselves.
- No changes are permitted to content of the International Release, except for the addition of new versions of this content in a module owned by the *terminology producer*. Any modifications resulting in changes to the classification of international content must be accompanied by a disclaimer notifying users of the differences between the extension edition and the International Edition.
- Any substantive improvements or corrections to the content in the International Edition that is made in an extension should be forwarded to SNOMED International in a timely fashion to improve the quality of the International Edition for all users.

c. Review and validation - SNOMED CT extension content should be validated using automated tests both at the time of authoring and before a release is packaged for distribution. In addition, manual review of the terminology content is also vital to ensure that it meets the quality standards for usability and clinical safety that can not be tested automatically.
d. Distribution - This involves classifying the associated SNOMED CT editions, packaging the release files, validating the distribution package, and making the release package available to *terminology consumers*.
e. Maintenance - A SNOMED CT edition must be maintained to respond to new change requests, and to ensure that the consistency and integrity with the international edition (and other modules on which it depends) is maintained appropriately.

It is important for extension producers to understand the complexity of authoring and managing an extension, and to prioritize the acquisition of appropriate tools to support the extension management processes. These tools are required to maintain the quality and integrity of SNOMED CT content, and to prevent errors which can easily be identified using automated processes. Extensions and editions should never be handcrafted or managed manually using a file-based approach (e.g. using spreadsheets). While a manual approach may be possible in the short term (given a solid understanding of the principles), version management for an evolving extension can quickly become unsustainable over time. In addition, a lack of automated validation can have serious patient safety implications.

2 Introduction

Background

SNOMED CT is a multilingual clinical terminology that covers a broad scope. However, some users may need additional concepts, relationships, descriptions or reference sets to support national, local or organizational needs.

SNOMED CT is designed to allow the International Edition to be enhanced by creating extensions to meet national or local requirements. The extension mechanism allows SNOMED CT to be customized to address the terminology needs of a country or organization that are not met by the International Edition. Extensions are managed by SNOMED International, and Members or Affiliate Licensees who have been issued a namespace identifier by SNOMED International. A namespace identifier is used to create globally unique SNOMED CT identifiers for each component (i.e. concept, description and relationship) within a Member or Affiliate extension. This ensures that references to extension concepts contained in health record data are unambiguous and can be clearly attributed to a specific issuing organization.

An extension may contain components and/or derivatives (e.g. reference sets used to represent subsets, maps or language preferences). Since the international edition and all extensions share a common structure, the same application software can be used to enter, store and process information from different extensions. Similarly, reference sets can be constructed to refer to content from both the international release and extensions. The common structure also makes it easier for content developed by an extension producer to be submitted for possible inclusion in a National Edition or the International Edition.

Purpose

The purpose of this guide is to be a practical starting point for extension producers. It is intended to help people who need to extend and configure SNOMED CT to meet national or local requirements for clinical documentation or particular business needs. It is important that extension producers are aware of the principles and processes involved in the creation and maintenance of extensions. Consistent approaches to the development and maintenance of extensions are required to avoid variation that may form a barrier to interoperability. The objective of this guide is therefore to explain the key use cases for extensions, to clarify the logical design of extensions, and to explain the key steps and principles for creating and maintaining extensions.

Audience

The primary audience of this guide is SNOMED International Members and other organizations that need to create, maintain and distribute SNOMED CT content using RF2.

We refer to these people and the organizations they work for as *terminology producers*, to distinguish them from *terminology consumers*, who use the SNOMED CT in system development, data capture, retrieval and analysis.

Although this guide is not targeted at *terminology consumers*, it contains some material that may be of interest to them, including:

- It explains the relationship between the international, national and local content that may form part of the SNOMED CT Edition that they use.
- It provides essential guidance for *terminology consumer* organizations that are considering whether to develop their own local extensions to configure SNOMED CT for their own use.

Document Overview

This document presents a practical guide to SNOMED CT extensions and is structured as follows:

- 1 Executive Summary presents the key messages from the document.
- 2 Introduction explains the background, purpose, scope, audience and overview of the document.
- 3 Purpose introduces the main purposes and key use cases for SNOMED CT extensions.
- 4 Logical Design describes the logical design of SNOMED CT extensions.

- 5 Key Steps describes the key steps involved with the creation, distribution and maintenance of an extension. As part of this, the section explains the key principles and best practices for authoring content in an extension.

3 Purpose

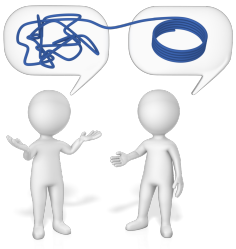
SNOMED CT is a multilingual clinical terminology that covers a broad scope of clinical concepts to an appropriate level of detail for international use. In addition, it provides an extension mechanism that allows SNOMED CT to be customized to address terminology needs that are not met by the International Edition. This mechanism enables new concepts, relationships, descriptions and reference sets to be added to support national or local needs.

However, not every new terminology requirement requires a SNOMED CT extension to be created. Potential extension producers should be aware of situations in which it is beneficial to create an extension, and situations where alternatives may be preferable. The following table provides a summary.

Create an extension when ...	An extension is not required when ...
• New national or local content is required to meet usage requirements; or • The content of the extension needs to be shared with other organizations; or • The standard file format and versioning mechanism of the release format 2 is needed	• No new national or local content is required; and • The content of the extension does not need to be shared with other organizations; and • Non-standard formats are sufficient to support local requirements.

When a SNOMED CT extension is required, the extension may serve a range of purposes. Successful extension producers must clearly understand the requirements and purpose of their extension, before deciding which types of SNOMED CT artifacts should be included. Table 3-1 below shows some typical purposes for a SNOMED CT extensions. For each purpose, the table includes some common use case examples and a list of SNOMED CT artifacts that should be included. For more information about each of these purposes, please click on the corresponding diagram.

Table 3-1: Purposes and use cases for extensions

Purpose	Use Case Examples	Included Artifacts
Translate SNOMED CT 	• Adding terms used in a local language or dialect • Adding terms used by a specific user group, such as patient friendly terms	• Descriptions • Language reference sets • Concepts (metadata only) • Relationships (metadata only)
Manage content gaps 	• Adding components that are missing in the International Edition • Adding concepts that are only relevant to a local context	• Concepts • Descriptions • Relationships • Language reference sets

Maps between SNOMED CT and other code systems 	• Maps between local codes and SNOMED CT • Maps between statistical classification systems and SNOMED CT	• Map reference sets • Concepts (metadata only) • Descriptions (metadata only) • Relationships (metadata only)
Configuration of the terminology for specific use cases 	• Specifying groups of components for reporting and analytics • Linking components to clinical knowledge resources	• Reference sets • Concepts (metadata only) • Descriptions (metadata only) • Relationships (metadata only)

3.1 Add Terms in Languages and Dialects

Overview

Representing language, dialect or specialty-specific term preferences is possible using a SNOMED CT extension. The logical design of SNOMED CT enables a single clinical idea to be associated with a range of terms or phrases from various languages, as depicted in Figure 3.1-1 below. In an extension, terms relevant for a particular country, speciality, hospital (or other organization) may be created, and different options for term preferences may be specified. Even within the same country, different regional dialects or specialty-specific languages exist may influence which synonyms are preferred. SNOMED CT supports this level of granularity for language preferences at the national or local level. Figure 3.1-2 illustrates how different terms, in different languages can be associated with the same concept.

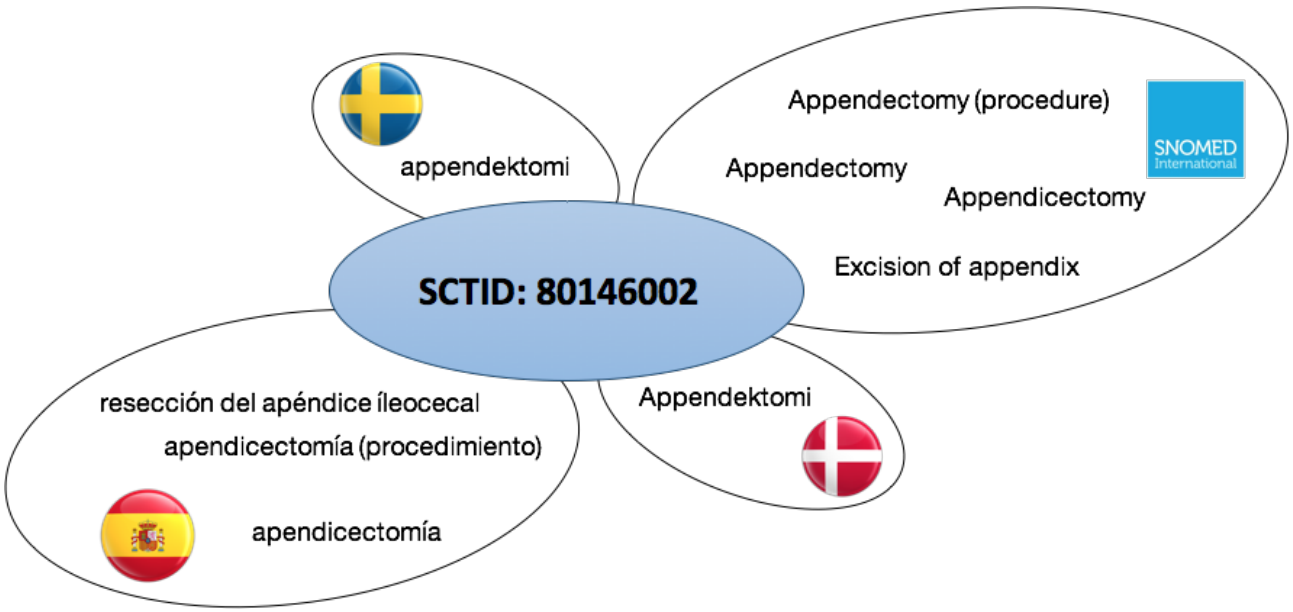


Figure 3.1-1: SNOMED CT allows a single clinical idea to be associated with a range of terms from various languages

Examples of situations in which additional terms may be required together with the type of organisation that may be involved in producing these terms.

What	Responsible
A translation of SNOMED CT into the language (-s) of the Member country/territory	National Release Center
Creation of patient-friendly terms	National Release Center, hospital or department, speciality
Creation of local synonyms	Vendor, hospital or department

⚠ Information on translating SNOMED CT can be found via the SNOMED International website, see translationinfo.

Key Steps

In section 5 Key Steps we explain the full set of steps for producing any extension. However, not all of these steps are required for all purposes. If producing an extension for the purpose of adding terms in languages and dialects, the following key steps are of particular relevance:

- **Create descriptions**
  - Principle and process for creating descriptions: 5.4.3.1 Add Description in an Extension
- **Specify the acceptability of the created descriptions in a language reference set**
  - Introduction to language reference set: 4.3.2.4.1 Language Reference Set
  - Principle and process for creating language reference set: 6.3.2.1 Create a New Reference Set
  - Principle and process for creating the individual reference set members: 6.3.3.1 Add Members to a Reference Set

3.2 Add Content to SNOMED CT

Extensions can be used to add content to SNOMED CT. These additions can address perceived content gaps in SNOMED CT because content is:

- Outside of the defined scope of the International Edition, and has thus been intentionally excluded
- Within the scope of the International Edition, but is not included as it relates to a new clinical knowledge or a previously unreported omission

In some cases, a content gap will be addressed by adding content directly to the International Edition, but in other cases the content will be rejected for inclusion in the International Edition. When content is outside the scope of the International edition, yet required to support national or local needs, this content may be added to an extension. For example

- Content relevant to an entire country may be relevant to include in a national extension, for example
  - Concepts used for reporting and quality metrics in a specific country or region
  - Representation of specific procedural concepts such as a technique that is used nationally for testing purposes
  - Concepts that are mandated by government agencies that would otherwise be excluded from the International release by editorial policy

- Content relevant to a specific hospital may be included in a local extension, for example
  - Local interface terms or patient-friendly terms, added as extension descriptions which describe existing SNOMED CT concepts.
  - Concepts representing specific hospital wards or beds
- Content relevant to a specific vendor may be included in a local extension, for example
  - System-specific interface terms or codes, added as extension descriptions which describe existing SNOMED CT concepts.

As illustrated in Figure 3.2-1 below, content can be added to SNOMED CT by inclusion in the International Edition or in relevant extensions. National extensions will add the content relevant for multiple Affiliates (or stakeholders) within a country or member territory, whereas highly specific content, which is only relevant for a specific vendor product or hospital may be added in a local extension.

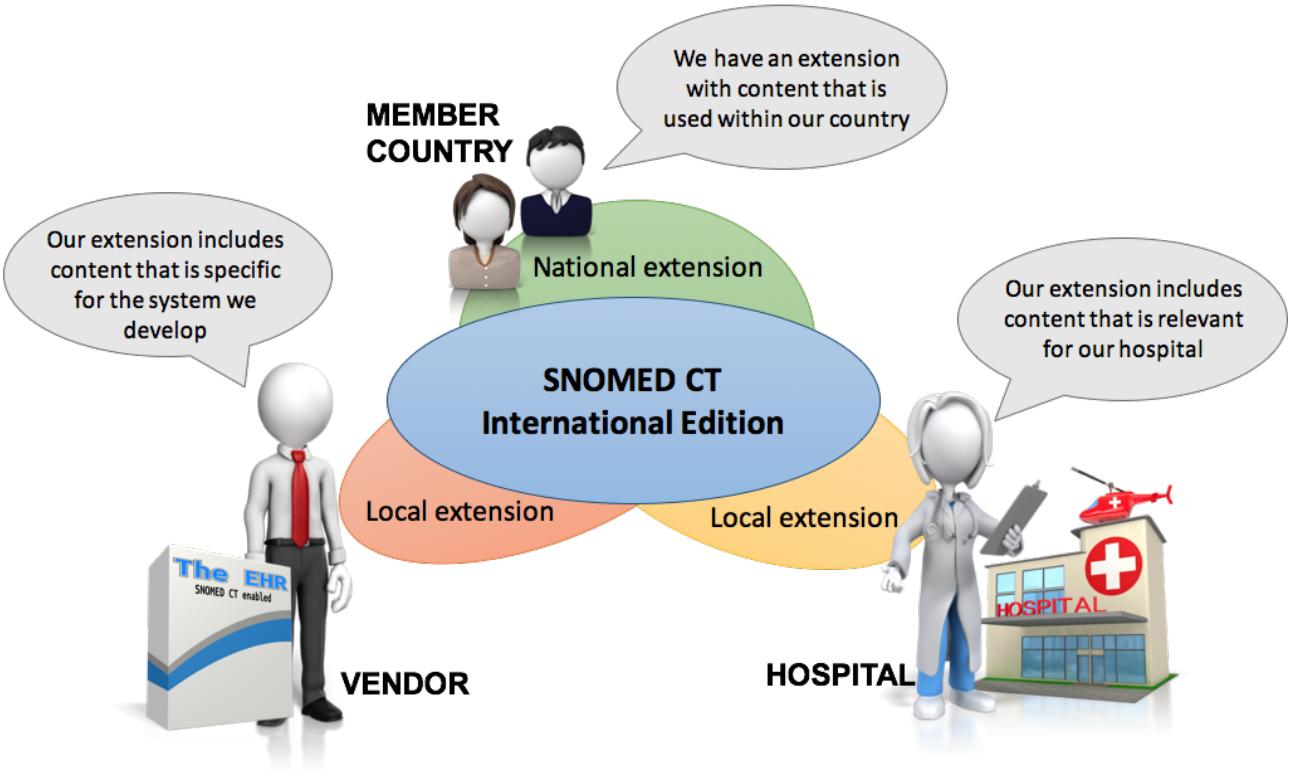


Figure 3.2-1: Extensions can be used to add national or local content to SNOMED CT

In some circumstances, extensions can be used to extend the expressivity of SNOMED CT, for example by expanding the concept model or by specifying additional terminology characteristics and features. SNOMED CT enables the following enhancements to expressivity to be made by adding terminology metadata.

- Creating a new reference set pattern, with new attributes
- Creating a new relationship type, i.e. a new attribute
- Creating a new description type

Once the required metadata has been added, the enhancement can be used within the extension and can be applied to SNOMED CT components in the extension, the International Edition, or any other modules on which the extension depends.

Extending the expressivity of SNOMED CT does not necessarily require the creation of new clinical content in SNOMED CT. However, it does require the addition of concepts within the |SNOMED CT Model Component| hierarchy of SNOMED CT and in some cases the addition of reference set members to one of the metadata reference sets.

The |SNOMED CT Model Component| hierarchy provides technical metadata supporting the SNOMED CT release. Before attempting to extend the expressivity of SNOMED CT, it is essential to understand how the metadata hierarchy is structured and how its concepts function.

Addition of metadata concepts can impact the integrity and consistency of SNOMED CT. For further information, please refer to the rules and advice on the addition of extension concepts to particular hierarchies in 5.3.2 Constraints on Concept Requests.

Key Steps

In section 5 Key Steps we explain the full set of steps for producing any extension. However, not all of these steps are required for all purposes. If producing an extension for the purpose adding content to SNOMED CT, the following key steps are of particular relevance:

- **Create components**
  - Principle and process for creating concepts: 5.4.2.1 Add Concept in an Extension
  - Principle and process for creating descriptions: 5.4.3.1 Add Description in an Extension
  - If you need to specify defining characteristics for concepts, for example following the creation of a new concept. See section 5.4.4.1 Add Relationship in an Extension for information about the principles and process for creating relationships
  - See the Practical Guide to Reference Sets for more information about the different reference set types and their usage

3.3 Map Between SNOMED CT and Other Code Systems

Overview

When implementing SNOMED CT there is often a need to link SNOMED CT to other code systems. Examples of practical uses of a map are described in the table below:

Purpose	Description
Integrating local codes and SNOMED CT	
Using a library of clinical phrases as an interface terminology	In some cases a library of clinical phrases, that is not part of SNOMED CT, may be used as the interface terminology for a particular data entry scenario. In this situation a map from the library of clinical phrases to SNOMED CT can support use of SNOMED CT for storage, analytics, and communication, while retaining the clinical phrases for presentation to the user.
Communication of clinical data between organizations	Messages and communication services are a means of exchanging data and thus enable effective and efficient sharing of information among healthcare professionals and between patients and providers. SNOMED CT is important for communication because it serves as a semantic foundation for the meaning expressed in a message. Therefore SNOMED CT can ensure consistent and accurate representation of the information communicated, and support the correct interpretation of the clinical information within a message. Mappings between SNOMED CT concepts and data elements within a communication or messaging specification may be developed and included in an extension to facilitate unambiguous communication of processable meaning across systems and organizations.
Migration to SNOMED CT	A map between the legacy codes from the original system and the corresponding SNOMED CT components can be developed to retain accessibility of legacy data as part of a system that uses SNOMED CT
Integrating statistical classification systems and SNOMED CT	

Purpose	Description
Statistical analysis of SNOMED CT encoded data	Clinical information recorded using SNOMED CT may include data that is relevant to reports, statistics, or billing claims which needs to be encoded using a specific code system or a statistical classification, such as ICD-10. Mapping from SNOMED CT to the relevant code system will allow information to be used for those purposes, and at the same time, minimize the requirement for additional manual data entry. Mapping also supports the idea that clinical information collected at the point of care should be represented in a semantically rich way that allows it to be retrieved and processed for a range of different purposes.
Meaning-based analysis of statistical data	Maps from other code systems or statistical classifications to SNOMED CT may also be included in an extension. For example, in situations where data is originally captured using a specific code system or statistical classification, but a map to SNOMED CT is required to enable analysis which takes advantage of the features of SNOMED CT, for example selective retrieval of data based on the defining characteristics of the concepts. For more information on this topic, please refer to section SNOMED CT Analytic Techniques, in Data Analytics with SNOMED CT.

Key Steps

In section 5 Key Steps we explain the full set of steps for producing any extension. However, not all of these steps are required for all purposes. If producing an extension for the purpose of representing a map between SNOMED CT and other code systems, the following key steps are of particular relevance:

- **Create the map reference set**
  - Principle and process for creating a reference set in an extension: 5.4.4.1 Create a New Reference Set in an Extension
- **Creating reference set members to represent each map record**
  - Principle and process for creating the individual reference set members: 5.4.5.1 Add Members to a Reference Set

3.4 Configure SNOMED CT for Specific Use Cases

The ability to tailor SNOMED CT to support specific use cases speaks to the adaptability of SNOMED CT, and any customization artifact that is beneficial to share may be useful to include in an extension. NRCs may need to configure SNOMED CT to match national requirements for reporting, analytics, language preferences etc.. Hence, following SNOMED CT reference sets may be included in an extension to meet a range of SNOMED CT use cases:

- A subset of SNOMED CT concepts or descriptions for use in a clinical speciality or practice
- Groups of concepts to support national reporting and analytics
- Prioritized components to support the logical display of search results
- Sets of components annotated with indicators that may trigger various guidelines or features
- Associations between SNOMED CT concepts, for example to represent language preferences or represent reasons for inactivation

In many cases, the terminology will be customized by referencing existing SNOMED CT components and linking these to additional information. The SNOMED CT reference set structure provides a standard representation for customized sets of SNOMED CT components. Different reference set types have been specified to meet particular needs and data structure requirements. All reference sets belong to a particular SNOMED CT Edition, and reference sets developed by Member countries or local organizations are included in their national or local extension.

Key Steps

In section 5 Key Steps we explain the full set of steps for producing any extension. However, not all of these steps are required for all purposes. If producing an extension for the purpose configuring SNOMED CT for a specific use case, the following key steps are of particular relevance:

- **Create the reference set**
  - Principle and process for creating a reference set in an extension: 5.4.4.1 Create a New Reference Set in an Extension

    - See the Practical Guide to Reference Sets for more information about the different reference set types and their usage
- **Creating reference set members to represent each map record**
    - Principle and process for creating the individual reference set members: 5.4.5.1 Add Members to a Reference Set

4 Logical Design

The logical design of a SNOMED CT extension is technically consistent with that of the International Edition. Both represent and version SNOMED CT components and reference sets in release files that conform to the Release Format 2 specification. This makes it easier for consumers of the International Edition to implement extensions. Figure 4-1 illustrates the logical design of a typical SNOMED CT local edition and its relationship to the associated local extension, National Edition and International Edition.

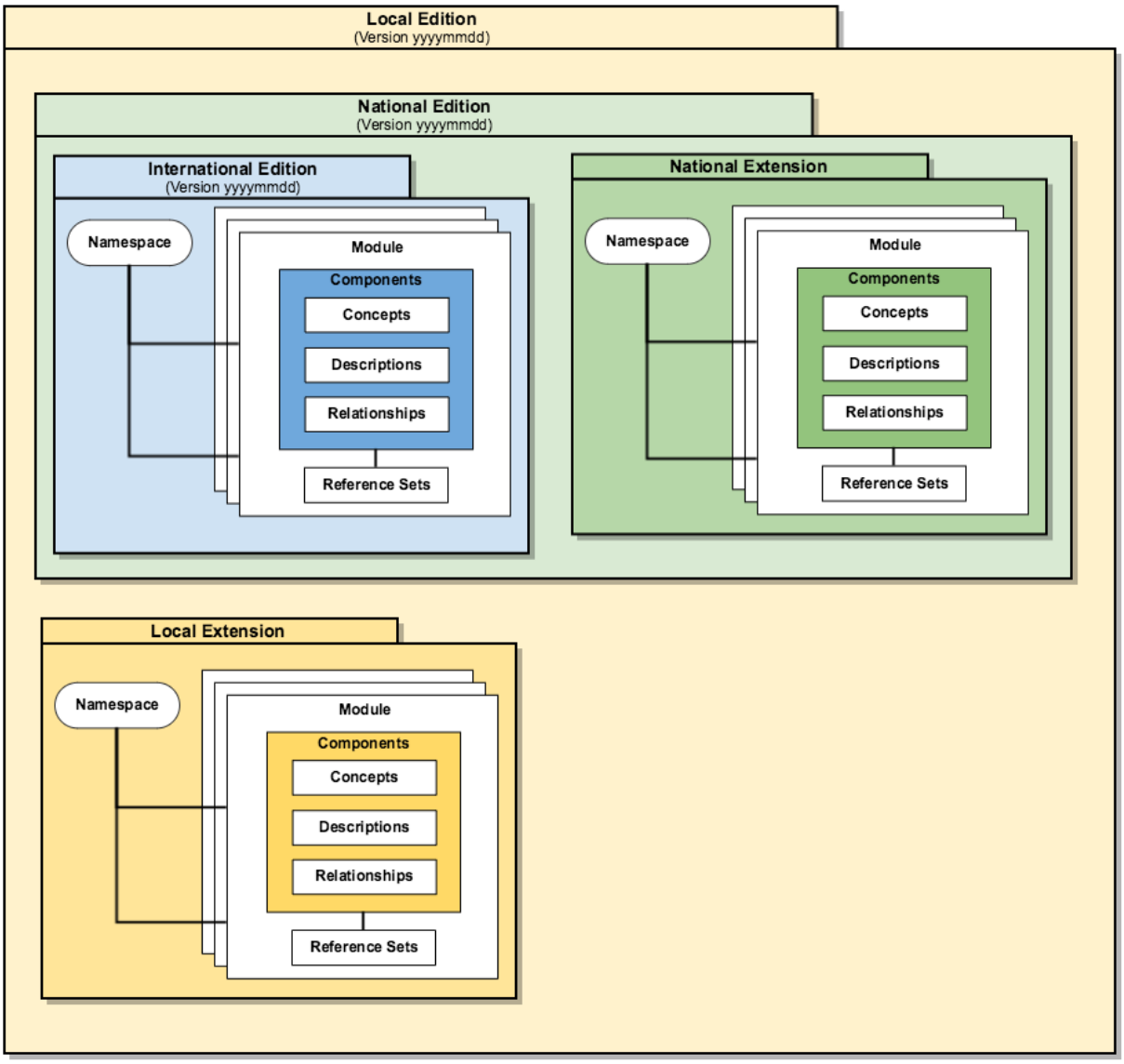


Figure 4-1: Logical design of a SNOMED CT local edition

Every SNOMED CT extension includes one or more modules, and each module contains either SNOMED CT components or reference sets (or both). Modules may be dependent on other modules. A national or local extension uses a namespace identifier issued by SNOMED International to ensure that all extension components can be uniquely identified (across all extensions). A SNOMED CT edition includes the contents of a focus module together with the contents of all the modules on which it depends. This includes the modules in the International Edition and possibly other modules from a national and/or local extension.

The following subsections explain this logical design in more detail:

4.1 Namespaces

The design of SNOMED CT specifies that all components (concepts, descriptions and relationships) are identified using SCTIDs. This is true for both components in the International Edition, and components in an extension. Extension producers therefore need a mechanism to ensure that their component identifiers are globally unique, and do not collide with the SCTIDs generated by other organizations. The SNOMED CT namespace identifier provides this mechanism. The namespace identifier ensures that all components within an extension are properly and uniquely identified, both locally and globally.

Prior to creating an extension, each organization must obtain a unique namespace identifier from SNOMED International (see http://snomed.org/changeadd). Once the namespace identifier is issued, SNOMED International authorizes the allocated organization to generate SCTIDs within their allocated namespace to identify extension components. Each SNOMED CT namespace identifier is 7 digits in length. By including this 7 digit identifier in the appropriate position within an SCTID, and maintaining locally unique item identifiers, global uniqueness of SCTIDs can be ensured.

> Note: Information about applying for a namespace identifier can be found on the SNOMED International website at http://snomed.org/changeadd

Examples

The table below lists some examples of namespace identifiers and the organizations to which they are allocated. These organizations include National Release Centers (or NRCs), local organizations, and commercial vendors. Note that this table is an excerpt from the SNOMED CT Namespace Registry as of October, 2017.

Table 4.1-1: Examples of namespace identifiers

Namespace ID	Organization
1000124	National Library of Medicine (USA) – IHTSDO Member
1000087	Canada Health Infoway (Canada) – IHTSDO Member
1000052	National Board of Health and Welfare (Sweden) – IHTSDO Member
1000119	Kaiser Permanente
1000129	B2i International LLC

Figure 4.1-1 below illustrates the structure of an extension SCTID using the concept 5281000124103 |Persistent asthma| from the 20170301 US Edition of SNOMED CT. The structure of the SCTID has been labeled and the namespace identifier is shown in red. Note that the spaces do not form part of the SCTID and have been added to improve readability.

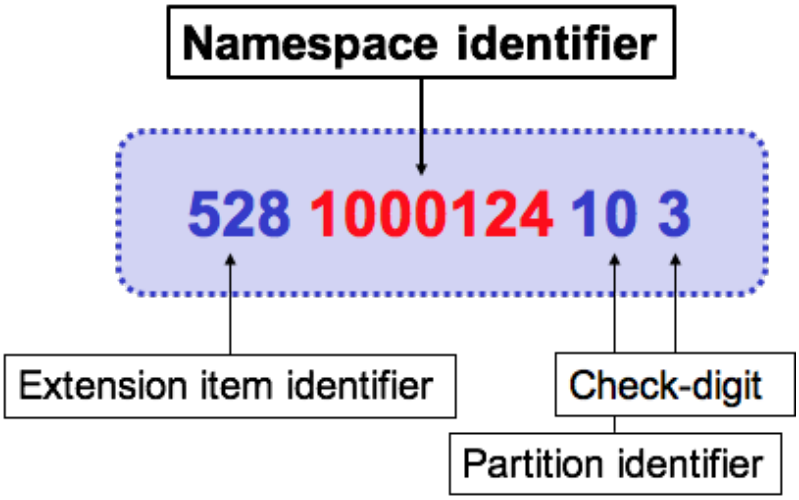


Figure 4.1-1: The structure of an example extension SCTID

Caution

The namespace identifier used in an extension's SCTIDs should NOT be used as the mechanism to determine which edition or extension a component belongs to. Instead, the **moduleId** attribute is used for this purpose. For more information, see 4.2 Modules.

For example, although 6811000087108 |CT guided biopsy of left lower limb| includes the namespace identifier "1000087" (allocated to Canada Health Infoway) and the partition identifier "10", this concept belongs to the |SNOMED CT core module| in the International Edition. The presence of the Canadian namespace identifier in this SCTID indicates that this concept was originally created by Canada Health Infoway and later promoted into the International Edition. For further information on content promotion, please refer to 5.7.1.3 Updating Extension.

4.2 Modules

In SNOMED CT, modules are used to organize content for maintenance and publication purposes. All concepts, descriptions, relationships, and reference set members must belong to a module. When a module is published, as part of a release package, all concepts, descriptions, relationships and reference sets that belong to that module must be published together. According to the logical design, this association between a component or reference set member and its associated module is made using the *moduleId* attribute. The moduleId attribute refers to a concept that represents and names the module in which a component or reference set is currently maintained. All components and reference set members within a module are maintained by a single organization.

SNOMED International Modules

The International Edition includes two modules. The core clinical components of SNOMED CT belong to the |SNOMED CT core module|. Metadata components, which support the specification of the terminology, belong to the |SNOMED CT model component module|. Both these modules are maintained by SNOMED International.

SNOMED International also maintains several other modules that supplement, rather that being part of, the International Edition. These include the |SNOMED CT to ICD-10 rule-based mapping module| and the |LOINC - SNOMED CT Cooperation Project module| .

Member and Affiliate Modules

Components and reference sets maintained by Members and Affiliate licensees are also organized into one or more modules. The module concepts used for this purpose must be created and maintained by the same organization. In most cases, the module concept and its associated descriptions and relationships will belong to the same module to which its identifier refers.[1]

Examples

Table 4.2-1 below lists some examples of modules, together with the organization responsible for maintaining and distributing the contents of the module. Note that the namespace identifier (highlighted in red) used in the module identifier refers to the organization who is responsible for that module.

Table 4.2-1: Examples of modules

Module Identifier	Module name	Maintained by
449080006	| SNOMED CT to ICD-10 rule-based mapping module |	SNOMED International
73 1000124 108	| US National Library of Medicine maintained module |	US National Library of Medicine – Member
2209 1000087 100	| Canada Health Infoway Reference Set Module |	Canada Health Infoway – Member
99900001 1000000 103	| SNOMED CT United Kingdom clinical extension module |	NHS Digital (UK) – Member

Table 4.2-2 includes a subset of columns and rows from the description file in the 20170301 US Edition. Note that the preferred term of the moduleId is included in the table for readability.

Table 4.2-2: Descriptions assigned to different modules in the US Edition

id	effectiveTime	active	moduleId	conceptId	term
301485011	20170731	1	900000000000207008 | SNOMED CT core module |	195967001	Asthma
1563 1000124 116	20120301	1	731000124108 | US National Library of Medicine maintained module |	52810001241 03	Persistent asthma
181114011	20170731	1	900000000000012004 | SNOMED CT model component module |	116680003	Is a

The example above reinforces some key points. Firstly, all components belong to exactly one module, as identified by the moduleId. Secondly, an edition may include components that belong to modules maintained by different organizations.

[1] The module concept may, alternatively, belong to a different module maintained by the same organization. When this is the case, the module will necessarily depend upon the module that contains its identifying concept. It is therefore usually simpler to keep the module concept within the module it identifies.

4.2.1 Module Definition

The first concept to be created in any extension is a module concept. The identifier of this module concept becomes the moduleId to which all extension content is assigned. The moduleId uses the namespace identifier allocated to the extension producer. Additional modules can be created within the same extension, if there is a requirement to maintain or publish sets of components separately. Module concepts must be created as a descendant of 900000000000443000 |Module (core metadata concept)| in a subhierarchy that is dedicated to the given extension provider. For more details about creating new concepts in an extension, please refer to 5.4.2.1 Add Concept in an Extension.

The first module concept in an extension, and its associated descriptions, relationships and language reference set members, must all belong to the given module. This means that the id of the new module concept will match the value assigned to moduleId for that row. Subsequent module concepts can either belong to its own module, or to another module owned by the same extension producer on which the given module depends. The module concept must have the definition status |Primitive| as the |SNOMED CT Model Component (metadata)| hierarchy has no concept model attributes. Please see the following page for information about Module Naming Conventions.

Example

The following example uses the module concept 45991000052106 |SNOMED CT Sweden NRC maintained module| from the Swedish extension. Note that this concept uses the namespace identifier assigned to the Swedish NRC - 1000052. As we can see in Table 4.2.1-1, the module identifier appears in the id column and in the moduleId column of the concept table. A value of |Primitive| is used for the definitionStatusId.

Table 4.2.1-1: Module concept in concept table

id	effectiveTime	active	moduleId	definitionStatusId
45991000052106	20121221	1	45991000052106	900000000000074008 |Primitive|

Table 4.2.1-2 shows the two necessary descriptions for the module concept in the description table. Note that the same namespace identifier is used as part of the description identifiers.

Table 4.2.1-2: Module descriptions in description table

id	effectiveTime	active	moduleId	conceptId	languageCode	typeId	term	caseSignificanceId
360431100005210	20121221	1	45991000052106	45991000052106	en	900000000000003001 |Fully specified name|	SNOMED CT Sweden NRC maintained module (core metadata concept)	900000000000448009 |Entire term case insensitive|
360432100005210	20121221	1	45991000052106	45991000052106	en	900000000000013009 |Synonym|	SNOMED CT Sweden NRC maintained module	900000000000448009 |Entire term case insensitive|

Table 4.2.1-3 shows the required |Is a| relationship for the module concept in the relationship table. Note that the same namespace identifier is used as part of the relationship identifier.

Table 4.2.1-3: Module relationship in relationship table

id	effectiveTime	active	moduleId	sourceId	destinationId	relationshipGroup	typeId	characteristicTypeId	modifierId

87210000052122	20121221	1	45991000052106	45991000052106	900000000000443000 |Module|	0	116680003 |Is a|	900000000000451002 | Existential restriction modifier|	900000000000011006 |Inferred relationship|

Note that a reference set is also used to specify the language preferences for the module concept. For additional details please refer to 4.3.2.4.1 Language Reference Set.

Please note that in the tables above the Preferred Term is shown for some fields of type SCTID to aid the readability of the tables.

4.2.2 Module Dependencies

Overview

Content in one module may refer to content in another module. For example, a concept in an extension module may by the source of an |is a| relationship, whose destination is a concept in the international core module. Content dependencies may include:

- Extension concepts with an |Is a| relationship whose destination is in another module
- Extension concepts with a defining attribute relationship whose destination is in another module
- Extension concepts with a defining relationship whose type belongs to another module
- Extension descriptions which are associated with a concept from another module
- Extension reference set members that reference components from another module
- Extension components with properties (e.g. definitionStatusId) that belong to the international |SNOMED CT model component module|

Figure 4.2.2-1 illustrates a local extension module that references content in a national extension, and is therefore dependent on the national extension module; and similarly a national extension module that references content in the International Edition, and is therefore dependent on the international modules.

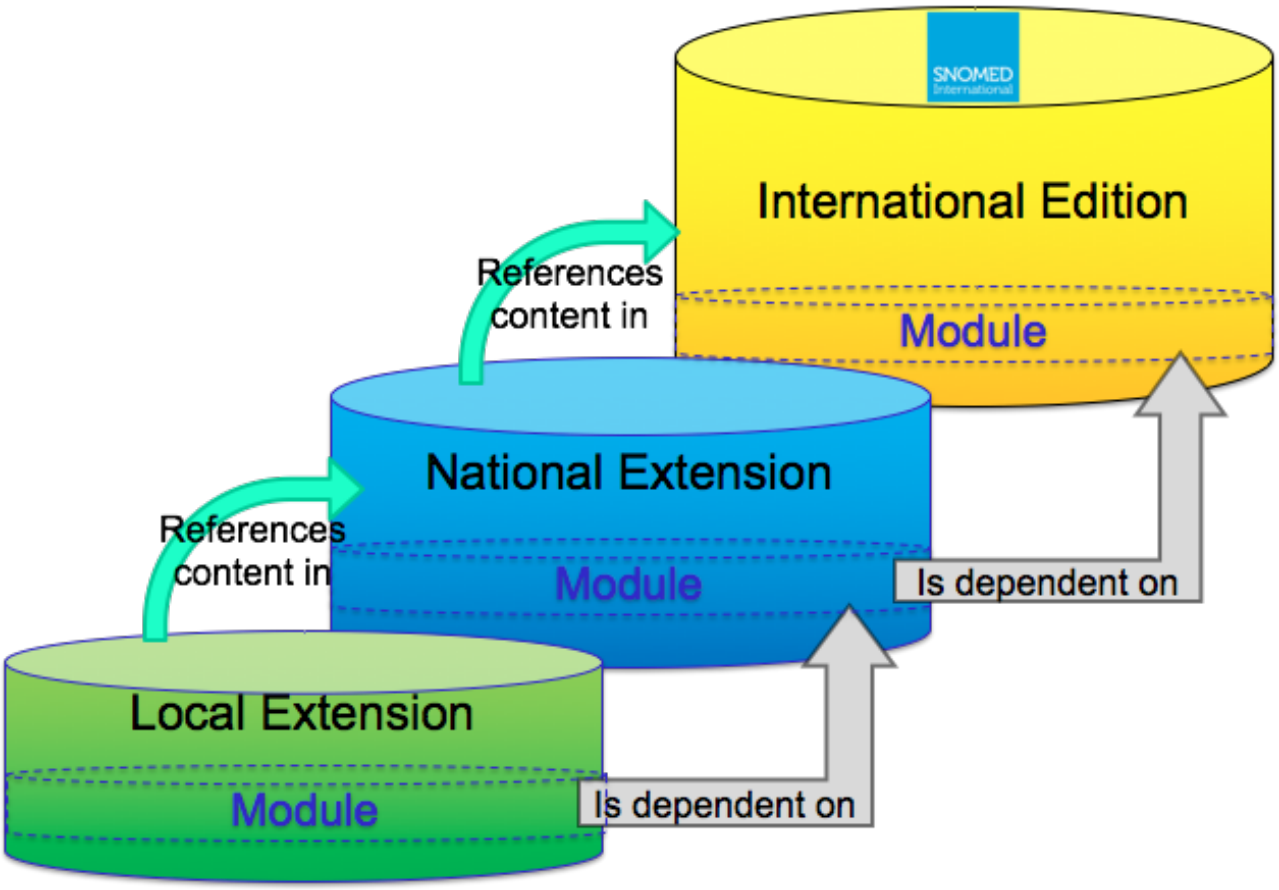


Figure 4.2.2-1: A dependent module references content in the module it depends on

In this situation, we refer to the local extension as being a 'child' of the national extension 'parent', because the local extension contains a module that is dependent on a module in the national extension. Similarly, the national extension is a 'child' of the International Edition 'parent', because the national extension contains a module that is dependent on a module from the International Edition.

Examples

The practical examples below demonstrate how content within a module can rely on another module, either directly or indirectly. As these examples also reference the logical design, the applicable SNOMED CT attributes have been **bolded**. For more detailed information on these attributes please refer to the SNOMED CT Release File Specifications.

Translation of Concepts

An extension, which includes translations of concepts from the International Edition, will include extension descriptions whose **conceptId** belongs to the International Edition. Therefore the content in this extension module directly relies on content in a module from the International Edition. As shown in Figure 4.2.2-2, this content dependency results in a module dependency.

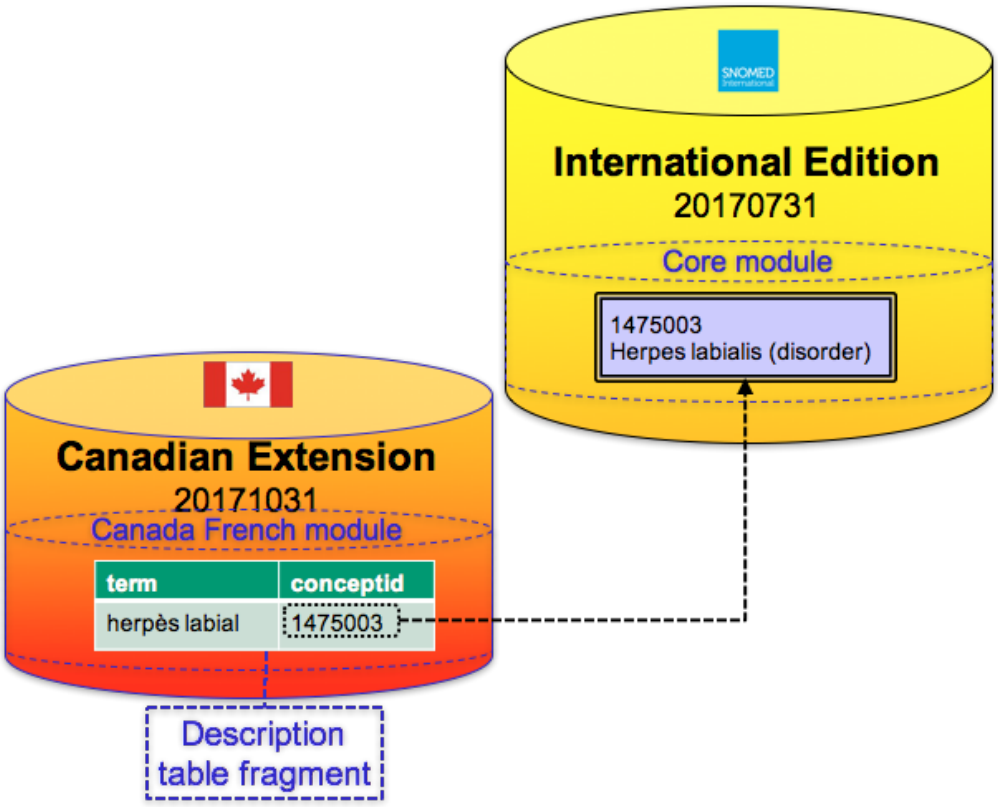


Figure 4.2.2-2: A content dependency caused by a translation

Defining Subsets of Concepts

In this example, a local extension contains a simple reference set that represents a subset of concepts. If one or more of the concepts in this subset belong to the National Extension, the extension reference set will contain rows in which the **referencedComponentId** refers to concepts from a module in the national extension. In this way, content from the local extension module relies on a module from the National Extension. Figure 4.2.2-6 illustrates another example in which a content dependency results in a module dependency.

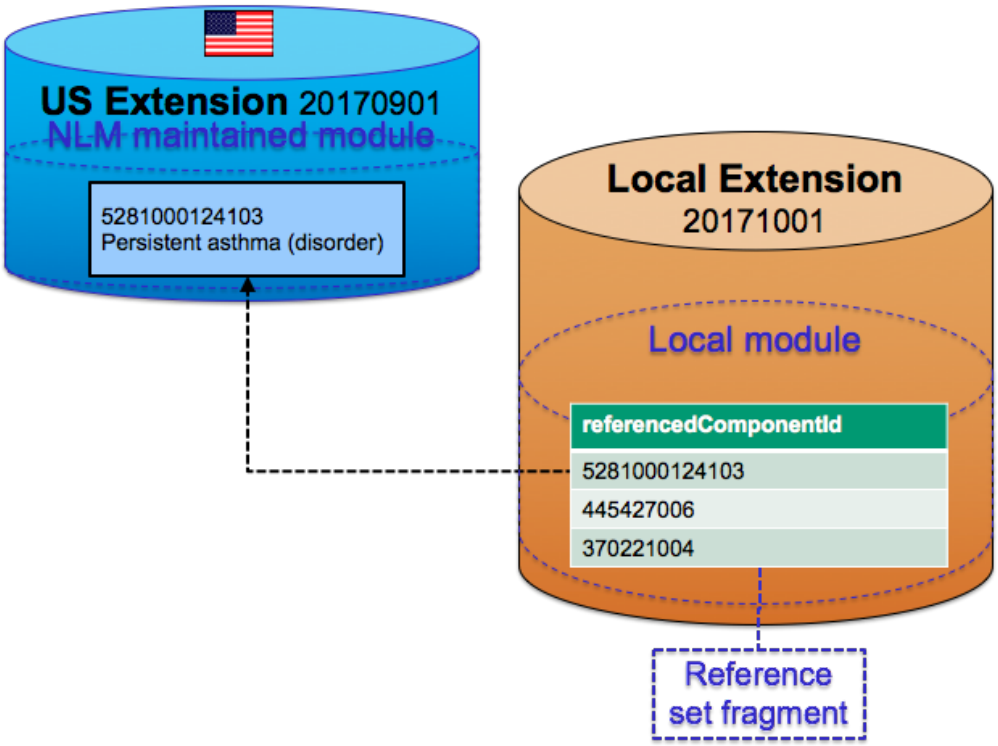
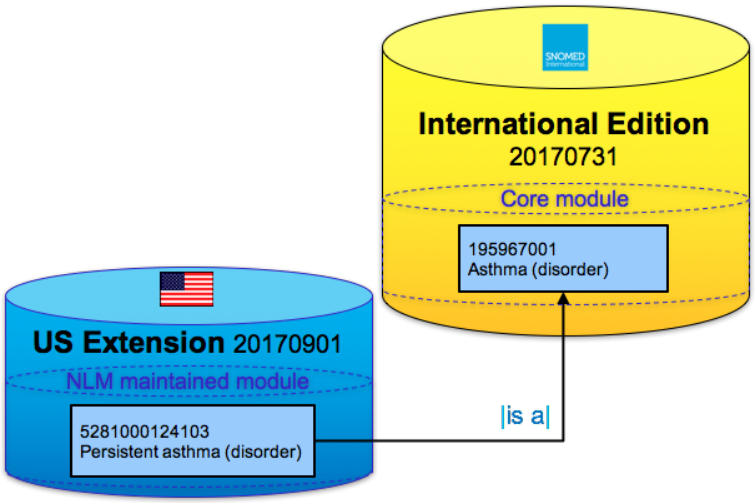


Figure 4.2.2-3: A content dependency caused by a reference set member

Adding Concepts

In this example, the extension includes new clinical content. Because all active concepts in an extension must be subsumed by the root concept, |SNOMED CT Concept|, there must exist at least one |is a| relationship in the extension with a **destinationId** that refers to a concept in another module. Clinical content in the extension module must depend on the international core module either directly or indirectly. Figure 4.2.2-6 illustrates another example in which a content dependency results in a module dependency.



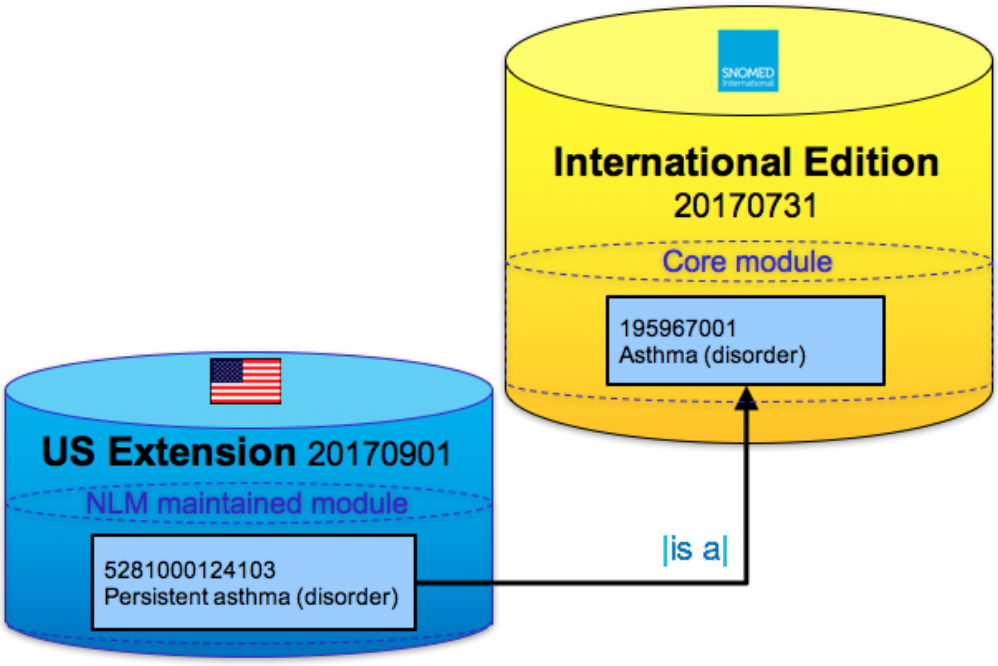


Figure 4.2.2-4: A content dependency caused by an |is a| relationship

Module Dependency Reference Set

The content dependencies described above create a need to establish a formal link between the dependent module and the modules on which it depends. These dependencies are defined in SNOMED CT at the module level, rather than being defined based on an entire extension. This allows extensions to be subdivided into separate modules, with different dependencies on modules in the same extension, in other extensions, or in the International Edition.

It is important to note that module dependencies are *version* specific. This means that each module dependency specifies the specific version of the source (child) module that is dependent on a specific version of the target (parent) module. This approach is important as new versions of either the source or target modules may affect the required dependencies.

A module is dependent on another module if it contains references to any component in another module.

A module may only contain references to SCTIDs which exist within the module itself, or in any module it depends on.

The extension producer is responsible for specifying the module dependencies for every version of each extension module. Module dependencies are specified by adding rows in the |Module dependency reference set|. For more information on this reference set format, please refer to 4.3.2.4.2 Module Dependency Reference Set.

It is important to note that all dependencies for each module (including transitive dependencies) must be explicitly stated. For example, Figure 4.2.2-5 illustrates an example in which all 3 dependencies must be explicitly stated: Module C is dependent on Module B, Module B is dependent on Module A and Module C is dependent on Module A.

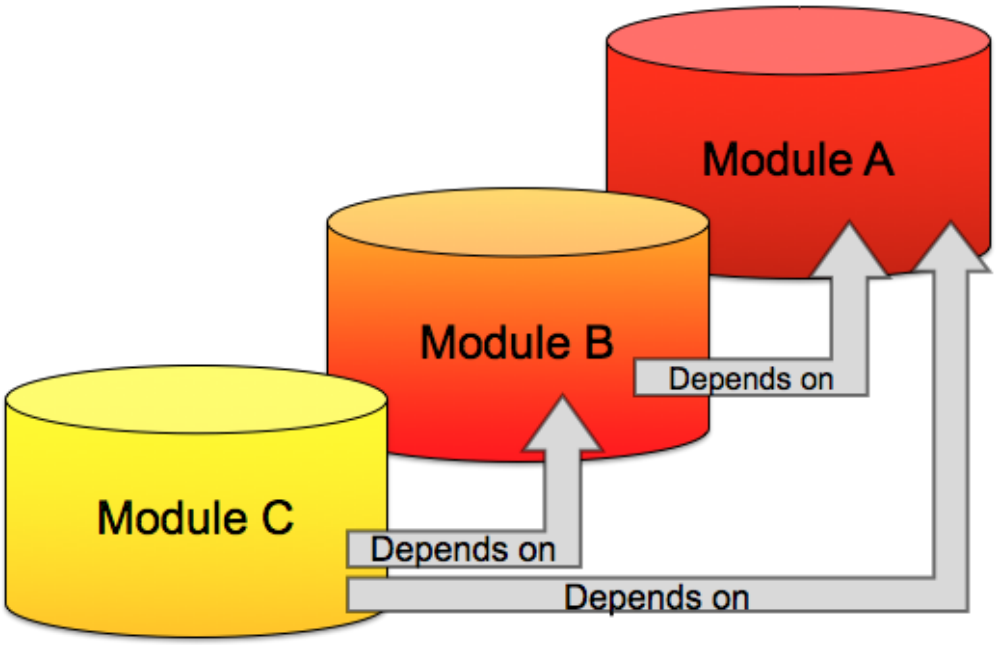


Figure 4.2.2-5: Explicitly stating transitive module dependencies

4.3 Extensions

A SNOMED CT extension is a set of components and reference set members that add to the SNOMED CT International Edition. An extension is created, structured, maintained and distributed in accordance with SNOMED CT specifications and guidelines. Unlike, the International Edition an extension is not a standalone terminology. The content in an extension depends on the SNOMED CT International Edition, and must be used together with the International Edition and any other extension module on which it depends.

Modules in an Extension

As explained in 4.2 Modules, every extension must contain at least one module, and all content in an extension must belong to one of its extension modules. More than one module can be created within the same extension, if there is a business requirement to maintain or publish sets of components separately. The set of module concepts that are maintained by the same extension producer are grouped together in a single module subhierarchy. Figure 4.3-1 below shows the module subhierarchy for the UK's National Release Center from the 20170401 UK Edition.

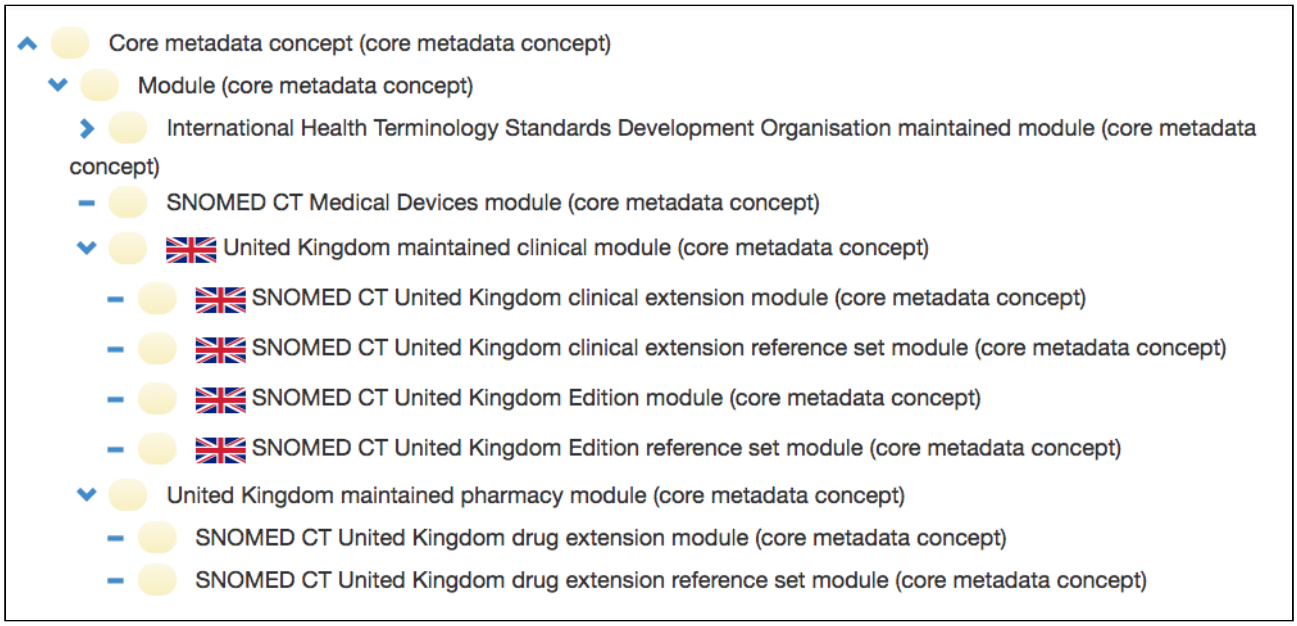


Figure 4.3-1: Module hierarchy from 20170401 UK Edition

Please note that some module concepts in the subhierarchy (e.g. |United Kingdom maintained clinical module (core metadata concept)|)serve as grouper concepts, which are never actually used as the moduleId of any content. The use of separate modules in the UK extension allow them to maintain and publish separate editions that include different content (e.g. with and without the drug extension). Using separate modules also allows module-based filters to be applied when searching content for a specific use case.

> Please nNote that the hierarchical structure shown in the image above **does not** represent the dependencies between modules. Dependencies are managed separately in the |Module dependency reference set|, as described in 4.2.2 Module Dependencies.

Extension Versions

A specific version of an extension can be referred to using the date on which the extension was published. Versioned extension modules can be identified using the SNOMED CT URI standard format: http://snomed.info/module/{sctid}/time/{timestamp}. For more information, please refer to 2.4 URIs for Modules and Versioned Editions.

4.3.1 Components

Most extensions include additional concepts, descriptions and/or relationships. It is therefore important for extension producers to understand the logical design of these components, and any extension-specific rule related to these components.

The following subsections present logical design rules and considerations for components in an extension:

For information about the process of authoring extension components, please refer to 5.4 Authoring. For a full list of RF2 component tables and their attributes, please refer to the SNOMED CT Release File Specifications.

4.3.1.1 Common Attributes

All components in an extension are represented using the same logical design as the International Edition. Concepts, descriptions and relationships all share four common attributes:

- id
- effectiveTime
- active
- moduleId

The following subsections explain how these attributes are used in an extension.

Id

The first attribute of every extension component is **'id'**. The id uniquely identifies the component across all extensions globally. It is therefore important that extension identifiers are structured according to agreed rules. Figure 4.3.1.1-1 shows the four parts of an extension SNOMED CT identifier (SCTID) - item identifier, namespace identifier, partition identifier and check digit.

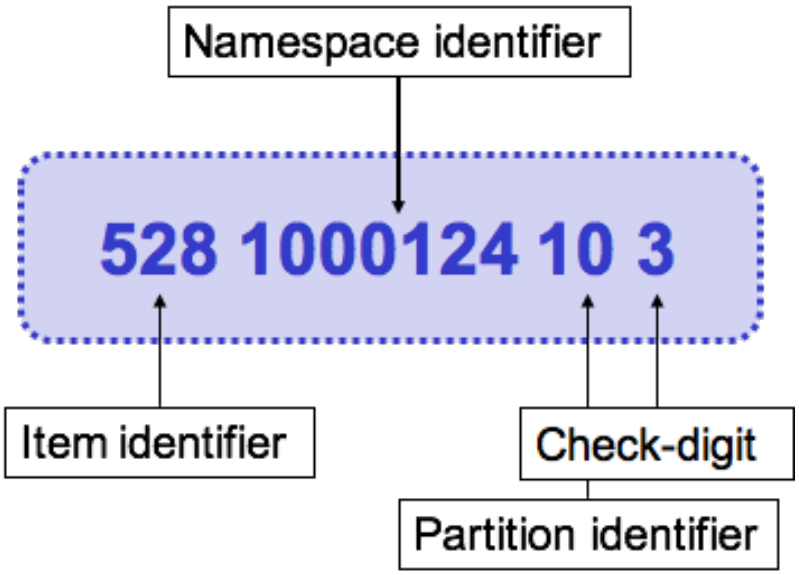


Figure 4.3.1.1-1: The four parts of an extension SCTID

Table 4.3.1.1-1 below explains the rules related to each of the four parts of a SNOMED CT identifier in an extension.

Table 4.3.1.1-1: Rules for each part of an extension SCTID

Order	Part	Rules	Example
1	Item identifier	• 1 to 8 digits in length • Allocated by the extension producer • Must be unique for the given namespace identifier and component type	528
2	namespace identifier	• 7 digits in length • Allocated by SNOMED International to the extension producer • Not used in components created by SNOMED International	1000124

3	partition identifier	• 2 digits in length
		• The following partition identifiers are used in an extension:
		  • 10 - extension concepts
		  • 11 - extension descriptions
		  • 12 - extension relationships
			10

4	check digit	• 1 digit in length • Computed from other digits in the SCTID using the Verhoeff check algorithm	3

effectiveTime

The second attribute of every extension component is **'effectiveTime**'. This attribute specifies the date on which the specific version of the component was released, using the ISO 8601 YYYYMMDD format. When a component is added or modified in an extension, the effectiveTime of the component version will match the version date of the release.

active

The third attribute of every extension component is '**active**'. This attribute specifies whether or not the specific version of the component is active at the given effectiveTime. The data type of this attribute is Boolean, with a

value of "1" indicating that the component is active at the given effectiveTime and a value of "0" indicating that the component is inactive at the given effectiveTime. New components in an extension will use an active value of "1".

moduleId

The fourth attribute in every extension component is '**moduleId**'. This attribute specifies the module in which the component is being maintained at the given effective time. The value of this attribute is the SCTID of a module concept that is created and maintained by the extension producer. Please refer to 4.2.1 Module Definition for more information on this topic.

4.3.1.2 Concepts

Overview

Concepts can be added to an extension, either to address content gaps or to expand the scope of SNOMED CT to include concepts that are used locally. Extension concepts are distributed in a concept file, which follows the standard RF2 release file format. For more details on the principles and processes for authoring concepts in an extension, please refer to 5.4.2 Authoring Concepts.

Attributes

In addition to the four common attributes, extension concepts also have a definitionStatusId.

definitionStatusId

The definitionStatusId attribute specifies whether the concept is primitive or fully defined. If defining relationships are added, which sufficiently define the concept, then the definitionStatusId should be set to |Defined|. Otherwise, the extension concept is |Primative|.

Example

In Figure 4.3.1.2-1 below, a row from the concept table of the 20170901 US edition is shown, which represents the concept 5281000124103 |Persistent asthma|.

Note that the attribute values in this row have been set according to the rules previously described.

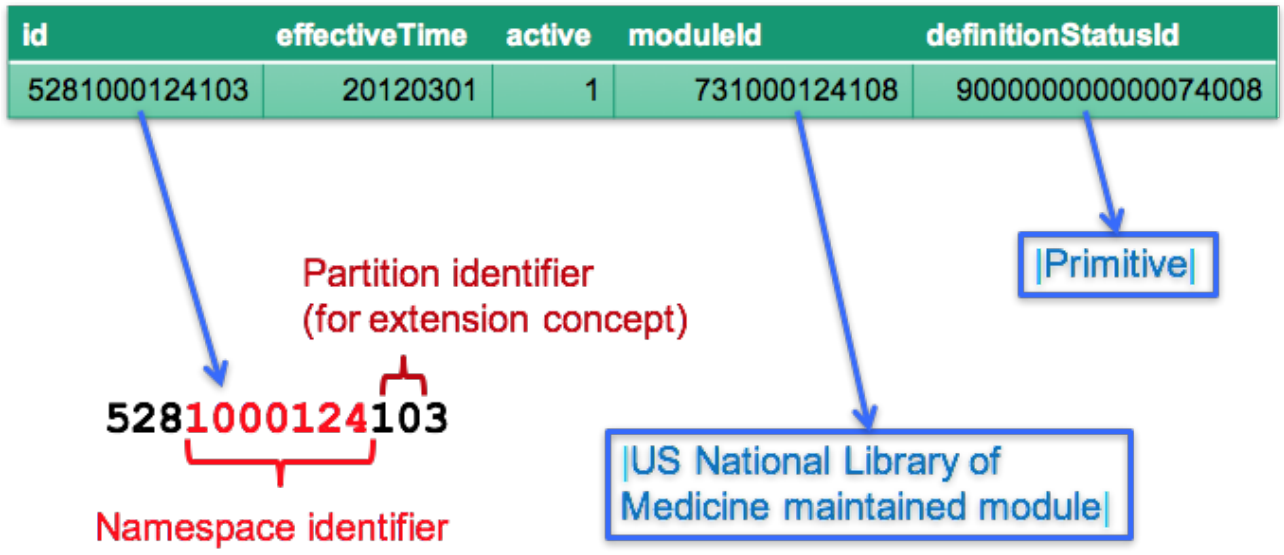


Figure 4.3.1.2-1: A row from the concept table in the 20170901 US Edition

Please note the following:

- The concept id uses a namespace identifier allocated to the National Library of Medicine (NLM). This indicates that this concept was originally created by the NLM.
- The concept id uses a partition identifier of “10”. This indicates that the concept was created in an extension.
- The moduleId indicates that this concept is included in the |US National Library of Medicine maintained module|, which is a module in the US Edition.
- The definitionStatusId states that this concept is |Primative|.

4.3.1.3 Descriptions

Overview

A description may be created in an extension as part of the process of adding a new concept, or adding a translation or alternative synonyms to an existing concept. Extension descriptions are distributed in a description file, which follows the standard RF2 release file format. For more details on the principles and processes for authoring descriptions in an extension, please refer to 5.4.3 Authoring Descriptions.

Attributes

In addition to the four common attributes, descriptions also have a conceptId, languageCode, typeId, term and caseSignificanceId.

conceptId

The conceptId attribute links the description to the concept that it describes. The conceptId may refer to a concept in the same extension module or to a concept in a module on which the extension module depends.

languageCode

The languageCode is a two-character string, which specifies the language of the description using the ISO 639-1 standard (e.g. 'EN', 'ES'). Please note that the languageCode specifies only the language of the description (e.g. English or Spanish) and does not indicate the dialect (e.g. US English or GB English). Dialect preferences are represented by language reference sets. Extensions may use a languageCode not used by the International Edition.

typeId

The typeId is a SCTID that specifies the type of the description. The value of this attribute must be a descendant of |Description type|. Possible values include |Fully specified name|, |Synonym|, and |Definition|. Note that |Definition| is used for text definitions, which are not included in the main description distribution files. Instead, descriptions of type |Definition| are distributed in a separate text definitions file.

term

The term attribute is a text string that describes the given concept. The term is represented using UTF-8, which allows non-English character sets to be used, such as the accented characters used in Nordic languages, Hindi characters and Chinese characters. The maximum length of a fully specified name or synonym is defined in the |Description format reference set| as 255 characters. Longer terms can be used for text definitions, which may be up to 4096 characters in length.

caseSignificanceId

The caseSignificanceId attribute is a SCTID which specifies how the case of the characters in the term may be changed when using the term. The values of this attribute always refer to descendants of the concept |Case significance|, and at the time of this publication, the three possible values are |Entire term case insensitive|, |Entire term case sensitive|,or |Only initial character case insensitive|.

Examples

In Figure 4.3.1.3-1 below, a row is presented from the descriptions table of the 20170901 US edition. Note that this description is associated with the extension concept from the previous example.

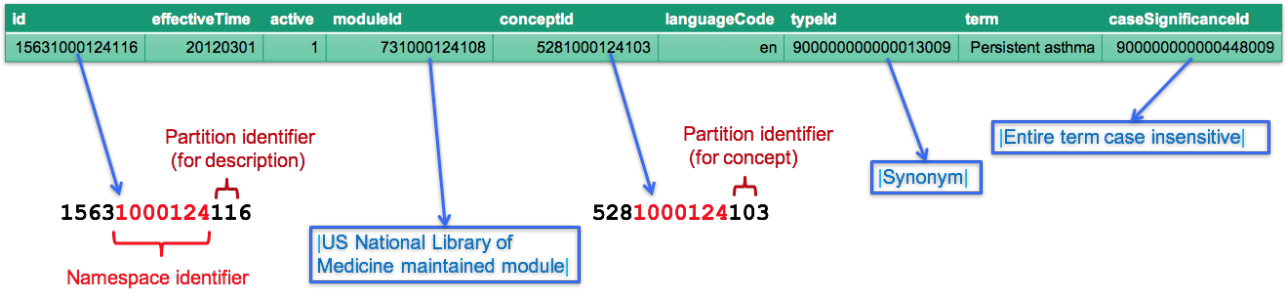


Figure 4.3.1.3-1: A row from the description table in the 20170901 US Edition

Please note the following:

- The description id uses a namespace identifier allocated to the National Library of Medicine (NLM). This indicates that this description was created by the NLM.
- The description id uses a partition identifier of “11”. This indicates that the description was created in an extension.
- The moduleId indicates that this description is included in the |US National Library of Medicine maintained module|, which is the same module that the associated concept (from the previous example) is included in.
- The description id and the conceptId both use a namespace identifier owned by the NLM. This indicates that this description is associated with a concept created by the same extension producer.
- A languageCode of "en" indicates that this description uses the English language.
- The typeId indicates that this description is a |Synonym|.
- The term indicates that the string "Persistent asthma" is used to describe the concept 5281000124103 .
- The caseSignificanceId indicates the term is |Entire term case insensitive|. This means that the term can be displayed using any alternative casing, such as "PERSISTENT ASTHMA" and "persistent asthma".

In Figure 4.3.1.3-2 below, a row is presented from the descriptions table of the 20170430 Canadian edition. Note that this description is assocaited with a concept from the International Edition.

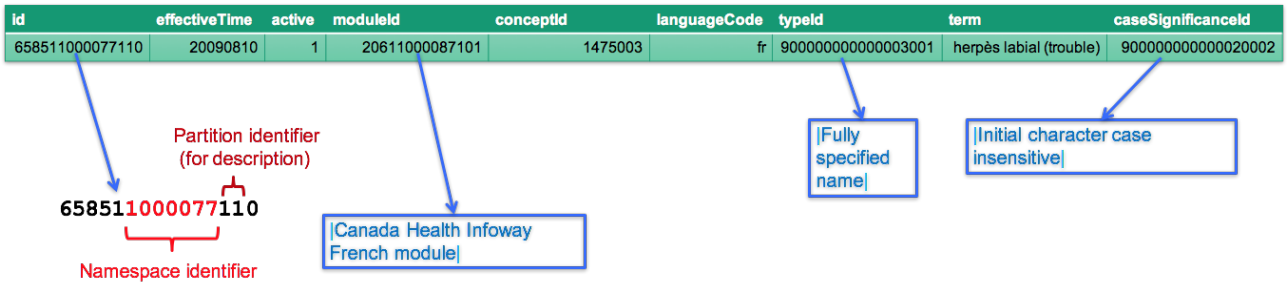


Figure 4.3.1.3-2: A row from the description table of the 20170401 French Canadian extension

Please note the following:

- The description id uses a namespace identifier allocated to Canada Health Infoway (CHI). This indicates that this description was created by the CHI.
- The description id uses a partition identifier of “11”. This indicates that the description was created in an extension.
- The moduleId indicates that this description is included in the |Canada Health Infoway French module|.
- The conceptId refers to the concept 1475003 |Herpes labialis| from the International Edition. This indicates that this term may have been created as a language specific term for an international concept.

- A languageCode of "fr" indicates that this description uses the French language.
- The typeId indicates that this description is a |Fully specified name|.
- The term indicates that the string "herpès labial (trouble)" is used to describe the concept 1475003 |Herpes labialis|. Please note the semantic tag, "trouble" is a French word for "disorder".
- The caseSignificanceId indicates the term is |Only initial character case insensitive|. This means that the term can be displayed as either "Herpès labial (trouble)" or "herpès labial (trouble)".

4.3.1.4 Relationships

Overview

Relationships are added in an extension to link a new concept to one of its supertypes, or to specify defining attribute relationships. Note that at least one hierarchical (|is a|) relationship is required for each new extension concept. Extension relationships are distributed in a relationship file, which follows the standard RF2 release file format. For more details on the principles and processes for authoring relationships in an extension, please refer to 5.4.4 Authoring Relationships.

Attributes

In addition to the four common attributes, relationships also have a sourceId, destinationId, relationshipGroup, typeId, characteristicTypeId and modifierId.

sourceId

The sourceId identifies the concept being defined by the relationship. In the case of a hierarchical (|is a|) relationship, the sourceId refers to the subtype/child concept.

destinationId

The destinationId represents the value of the defining relationship. In the case of a hierarchical (|is a|) relationship, the destinationId refers to the supertype/parent concept. The destinationId may refer to a concept in the same extension module as the source concept, or any module on which this extension module depends.

relationshipGroup

The relationshipGroup is an integer that indicates whether or not the relationship is grouped together with other relationships associated with the same source concept. Any relationships with the same sourceId and relationshipGroup integer (other than zero) are considered to belong to the same relationship group. A relationshipGroup of zero (i.e. "0") indicates that the relationship was not grouped with other relationships by the terminology author. For more information on relationship groups, please refer to relationship groups in SNOMED CT.

typeId

The typeId is a SCTID that specifies the attribute type of the relationship. In the case of a hierarchical relationship, the value of this attribute is |Is a|. Attribute relationships use a typeId value that refers to a subtype of |Concept model attribute|. Common examples of attribute types include |Finding site| and |Method|.

In most cases, the typeId will refer to an international concept. However, additional attribute concepts may be added in an extension if there is a legitimate clinical need, and clear rules and guidance is provided. For more information, please refer to 5.4.4.1 Add Relationship in an Extension.

characteristicTypeId

The characteristicTypeId is a SCTID, which specifies whether this relationship was stated by an author or was inferred by a description logic classifier. For more information, please refer to 5.6.1.1 Classifying an Edition.

modifierId

The modifierId attribute is a SCTID which specifies the type of description logic that applies to the given relationship - existential restriction (i.e. 'there exists some') or universal restriction (i.e. 'there exists only'). All relationships in the International Edition use the value |Existential restriction modifier|. This value is also recommended for extension relationships, as the value |Universal restriction modifier| can significantly affect the speed of classification.

Example

In Figure 4.3.1.4-1 below, two rows from the relationship table of the 20170901 US edition are shown. Note that these relationships are used to define the extension concept from the earlier example.

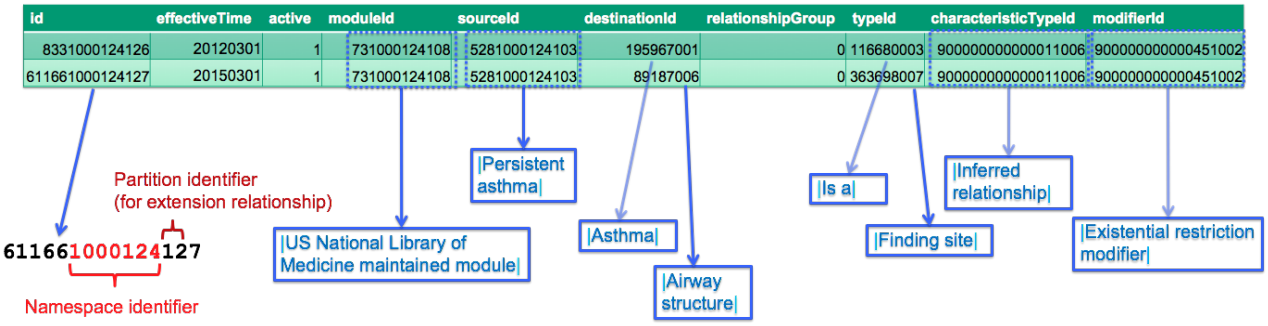


Figure 4.3.1.4-1: Rows from the relationship table in the 20170901 US Edition

Please note the following:

- The relationship ids use a namespace identifier allocated to the National Library of Medicine (NLM). This indicates that these relationships were originally created by the NLM.
- The relationship ids use a partition identifier of “12”. This indicates that the relationships were created in an extension.
- The moduleId indicates that these relationships are included in the |US National Library of Medicine maintained module|. This is the same module as the source concept belongs to.
- The sourceId indicates that the relationship defines the concept |Persistent asthma|.
- The typeId indicates that the first relationship is an |Is a| relationship, while the second relationship specifies the |Finding site| of the source concept.
- The destinationId indicates that |Asthma| is a supertype of |Is a|, and that the |Finding site| of |Is a| is |Airway structure|.

4.3.2 Reference Sets

It is important that extension producers understand how reference sets and their members are created. All extensions must include new members in the |Module dependency reference set| to define the modules on which the extension module(s) depend. Many extensions will also specify language or dialect preferences in a 900000000000506000 |Language type reference set|. Some extensions may also include additional reference sets to extend the use of SNOMED CT. For example, a subset of concepts for use in a specific clinical setting can be distributed as a |Simple type reference set|.

The following subsections present the logical design rules and considerations for reference sets and reference set members in an extension:

4.3.2.1 Common Attributes

Reference sets use a range of different attributes depending on the reference set type. While a detailed discussion of all reference set attributes is outside the scope of this document, this section briefly reviews some of the key attributes. Please refer to Reference Set Types in the Practical Guide to Reference Sets for more information on reference set types and their specific attributes. Please refer to 5.4.5 Authoring Reference Sets and 5.4.6 Authoring Reference Set Members for more information on the principles and processes for authoring reference sets and their members.

All reference sets share six common attributes:

- id
- effectiveTime
- active
- moduleId
- refsetId
- referencedComponentId

The following subsections explain how these attributes are used in an extension.

id

The first attribute of every extension reference set member is '**id**'. Unlike the id of a SNOMED CT component, the id of a reference set member is a UUID (Universally Unique Identifier). UUIDs are 128-bit unsigned integers, which are uniquely generated using widely available algorithms. This avoids the need to track the issuing of SCTIDs for thousands of reference set rows. When used in a reference set, the id identifies the reference set member or row.

Table 4.3.2.1-1: Example UUIDs

a07ea203-8db3-4a54-ae91-36bc20a766d2
55aafce9-981e-48e9-b4d0-8f542150cec7
f1c72bdf-0d98-493f-8f6e-dd799f1935df
3b0cc397-c99d-4cd5-90ba-7edaa7ea8771

In accordance with the versioning mechanism of SNOMED CT, a previously used UUID will be referenced, if and when there is a need to update the corresponding reference set row.

effectiveTime

The second attribute of every extension reference set member is **'effectiveTime**'. This attribute specifies the date on which the specific version of the reference set member was released, using the ISO 8601 YYYYMMDD format. When a reference set member is added or modified in an extension, the effectiveTime of the member version will match the version date of the release.

active

The third attribute of every extension reference set member is '**active**'. This attribute specifies whether or not the specific version of the reference set member is active at the given effectiveTime. The data type of this attribute is Boolean, with a value of "1" indicating that the member is active at the given effective time and a value of "0" indicating that the member is inactive at the given effectiveTime. New reference set members in an extension will use an active value of "1".

moduleId

The fourth attribute in every extension reference set member is '**moduleId**'. This attribute specifies the module in which the member is being maintained at the given effective time. The value of this attribute is the SCTID of a module concept that is created and maintained by the extension producer. Please refer to 4.2.1 Module Definition for more information on this topic.

refsetId

The fifth attribute in every extension reference set member is '**refsetId**'. This attribute refers to the id of the reference set concept that is used to represent the meaning of the reference set to which the given member belongs. The reference set concept must be a subtype of |Reference set|. The descriptions associated with this concept help to name the reference set. For more information, please refer to 4.3.2.2 Reference Set Example.

referencedComponentId

The sixth attribute of every extension reference set member is '**referencedComponentId**'. This attribute uses a SCTID to identify the component that is referenced by this member of the reference set. For example, the referencedComponentId may refer to a member of a subset represented by a |Simple type reference set|, to the description whose acceptability is being defined in a |Language type reference set|, or to the source of map in a |Simple map type reference set| ..

4.3.2.2 Reference Set Example

All reference sets require a concept to be created that represents the meaning of the reference set, provides the identification of the reference set, and names the reference set. These reference set concepts are created in the |Reference set (foundation metadata concept)| subhierarchy as subtypes of a reference set concept that represents

the associated reference set pattern. For more information about creating a reference set concept in an extension, please refer to 5.4.5.1 Create New Reference Set in an Extension.

In this section, we discuss an example reference set concept (|Route of administration reference set|) from the 20170430 Canadian extension, and explain how this concept and its associated descriptions, relationships and language preferences are represented.

Reference Set Concept

In Figure 4.3.2.2-1 below, a row from the concept table is shown for the concept |Route of administration reference set|.

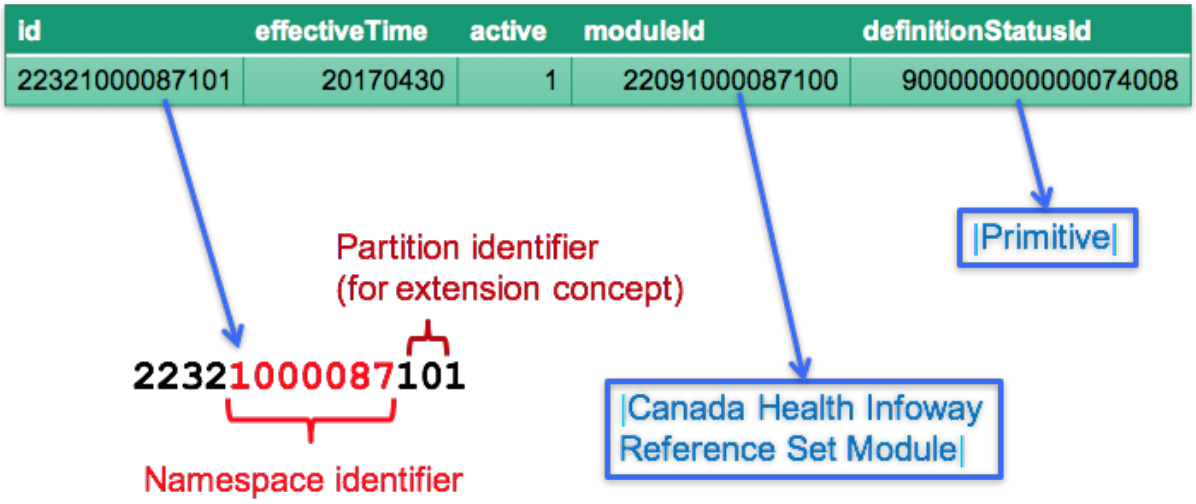


Figure 4.3.2.2-1: Example reference set concept in the concept table

Please note the following:

- The concept id uses a namespace identifier allocated to Canada Health Infoway.
- The concept id uses a partition identifier of "10" to indicate that this concept was create in an extension.
- The moduleId indicates that this concept belongs to the |Canada Health Infoway Reference Set Module|.
- The definitionStatusId indicates that this concept is |Primative|. Please note that all metadata concepts in SNOMED CT are |Primative|.

Reference Set Description

In Figure 4.3.2.2-2 below, a set of rows from the description table are shown. These descriptions provide terms that describe the concept |Route of administration reference set|.

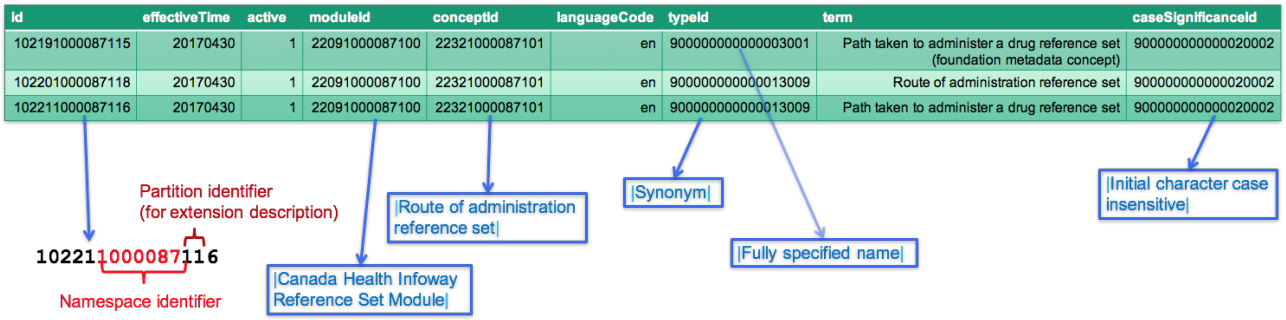


Figure 4.3.2.2-2: Example descriptions for the reference set concept

Please note the following:

- The description id uses a namespace identifier allocated to Canada Health Infoway.
- The description id uses a partition identifier of “11” to indicate that these descriptions were created in an extension.
- The moduleId indicates that these descriptions belong to the |Canada Health Infoway Reference Set Module| .
- The conceptId links the descriptions to the reference set concept |Route of administration reference set|.
- A languageCode of "en" indicates that the terms are written in English.
- The typeId indicates that the first description is a |Fully specified name|, while the following two are of type |Synonym|.
- The term is used to name the associated reference set.
- The caseSignificanceId indicates that the terms is |Only initial character case insensitive|.

Reference Set Relationship

In Figure 4.3.2.2-3, a row from the relationship table is shown. This relationship defines the supertype concept of |Route of administration reference set|.

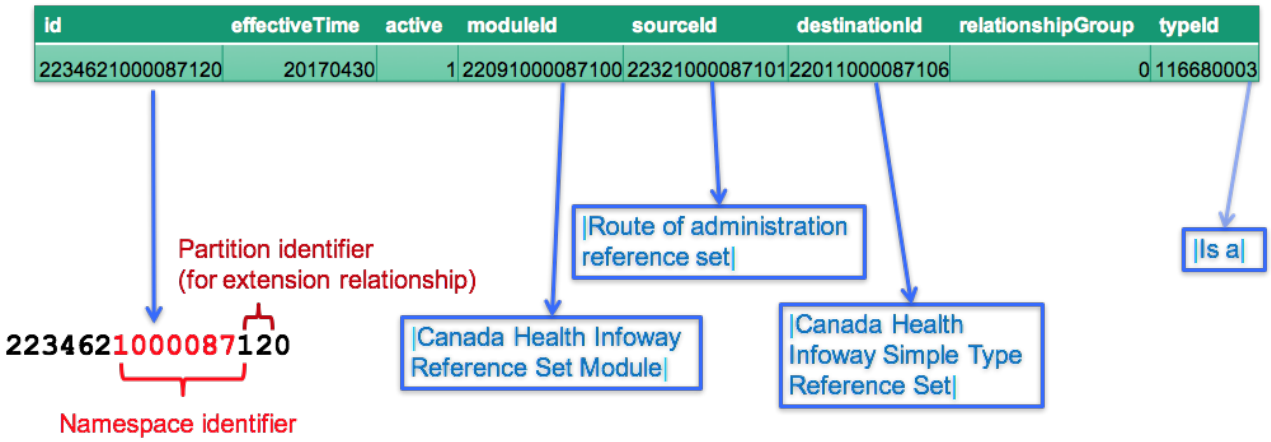


Figure 4.3.2.2-3: Example relationships for the reference set concept

Please note the following:

- The relationship id uses a namespace identifier allocated to Canada Health Infoway.
- The relationship id uses a partition identifier of “12” to indicate that this relationship was originally created in an extension.
- The moduleId indicates that these relationships belong to the |Canada Health Infoway Reference Set Module |.
- The sourceId indicates that the relationship applies to the concept |Route of administration reference set|.
- The typeId indicates that the relationship represents an |Is a| relationship.
- The destinationId indicates that |Canada Health Infoway Simple Type Reference Set| is a supertype of |Is a|.
- The characteristicTypeId and modifierId columns have been omitted from this diagram for brevity.

Reference Set Language preferences

In Figure 4.3.2.2-4 below, a set of rows from the |Canada English language reference set| are shown. These rows define the language preferences for the descriptions associated with the concept |Route of administration reference set|.

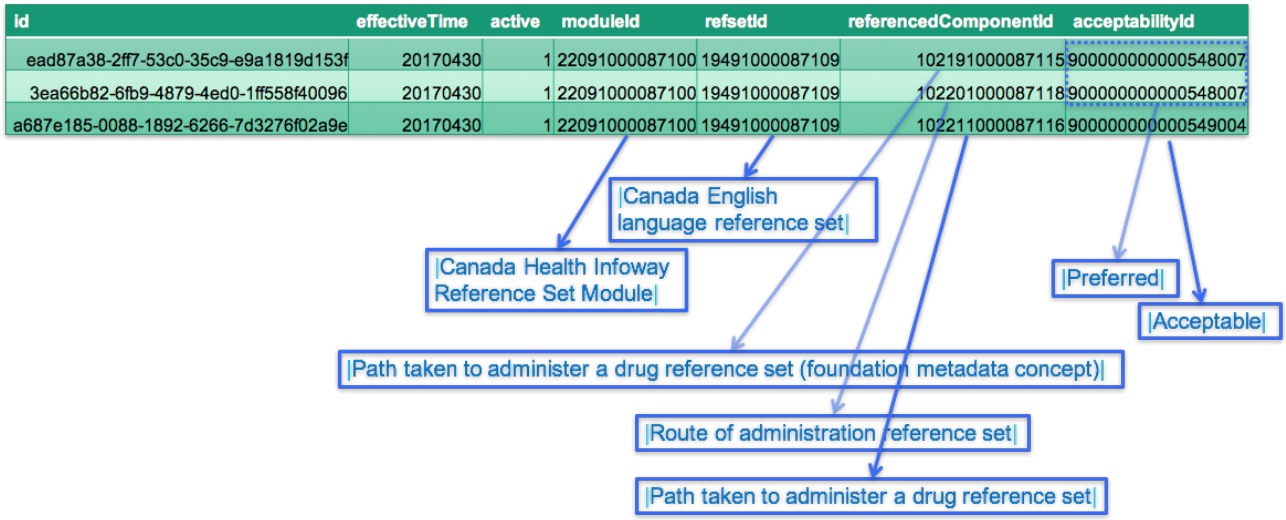


Figure 4.3.2.2-4: Example language preferences for the reference set concept

Please note the following:

- The id is a UUID that was generated for the corresponding refset member.
- The moduleId indicates that these refset members belong to the |Canada Health Infoway Reference Set Module|.
- The refsetId column indicates that these rows are members of the |Canada English language reference set|. Note, that if the relevant language reference set does not exist, this must be created using the same process.
- The referencedComponentId column refers to the descriptions for which the acceptability is being specified.
- The acceptabilityId column indicates the |Preferred| and |Acceptable| descriptions in the given language dialect (in this case Canadian English).

4.3.2.3 Reference Set Member Example

Once a concept exists, which identifies the reference set being populated, the individual reference set members can be added as rows to the associated reference set file.

In this section, we discuss an example reference set, and explain how its members are represented in an extension.

Simple Reference Set

The table below shows a set of rows from a |Simple type reference set| in the 20170430 Canadian extension.

Table 4.3.2.3-1: Fragment of the |Route of administration reference set| from the 20170430 Canadian extension

id	effectiveTime	active	moduleId	refsetId	referencedComponentId
0143c909-15a3-824b-e6fe-7498d915747a	20170430	1	22091000087100	22321000087101	372474001
0182cbb9-7925-4caf-6741-de80ae61055b	20170430	1	22091000087100	22321000087101	372464004
0346aebb-e051-c2bb-06e4-6e1627332433	20170430	1	22091000087100	22321000087101	417985001
105a2927-de95-ad8f-541c-39c7f752a07e	20170430	1	22091000087100	22321000087101	372449004
1516654e-d09b-404a-3b9b-39976a9c6753	20170430	1	22091000087100	22321000087101	418418000
15ddfd66-abc6-9752-1183-21d37e88c083	20170430	1	22091000087100	22321000087101	445768003
1aebaa63-0f1d-e8d6-ef51-4c64fe2da26f	20170430	1	22091000087100	22321000087101	419684008
1ba55d9a-9bd8-a6f5-e148-9f270774197d	20170430	1	22091000087100	22321000087101	372450004

id	effectiveTime	active	moduleId	refsetId	referencedComponentId
1fa8f320-2445-2b1d-ee33-a6af6810af07	20170430	1	22091000087100	22321000087101	445769006
26b886ee-7773-48e9-7fff-36d062f6ac30	20170430	1	22091000087100	22321000087101	37161004
281841c9-657e-273f-e569-6892e7c16ceb	20170430	1	22091000087100	22321000087101	418586008
28f12275-fdff-b178-05e9-b48e8929d45a	20170430	1	22091000087100	22321000087101	461657851000087101

Please note the following:

- All reference set members have a unique UUID.
- The effectiveTime and active values indicate that these rows were active from 20170430.
- The moduleId indicates that these rows belong to the |Canada Health Infoway Reference Set Module|.
- The refsetId indicates that each member belongs to the |Route of administration reference set|.
- The referencedComponentId column refers to the concept that the given reference set member is referencing. Note that most of the referenced components are international concepts, while the referencedComponentId in the last row refers to a concept created in the Canadian extension. Please also note that the version of the referenced component that is referred to is established according to the dependencies defined in the |Module dependency reference set|.

4.3.2.4 Essential Reference Sets

All extensions must specify their extension module dependencies and language preferences. In this section, we describe the essential reference sets that support this.

4.3.2.4.1 Language Reference Set

Whenever a new description is added in an extension, the language preferences associated with that description must be specified. This is done by adding rows to a language reference set to indicate whether each description is |acceptable| or |preferred| in the given language or dialect.

Attributes

As previously mentioned, all reference sets use six common attributes. In addition to these, a |Language type reference set| also includes an **acceptabilityId**. The table below explains how each of these attributes is populated in a |Language type reference set|.

Table 4.3.2.4.1-1: Summary of language reference set attributes

Attribute	Data Type	Use in a Language Reference Set
id	UUID	Can be generated using a standard UUID generator.
effectiveTime	Time	Specifies the date on which this version of the reference set member was released.
active	Boolean	Indicates whether or not the reference set member is active at the given effectiveTime. This value is set to "1" for new reference set members.
moduleId	SCTID	Identifies the module to which this reference set member belongs. For members in an extension, this will always be a module created by the extension producer.
refsetId	SCTID	This refers to the concept id of a descendant of |Language type reference set| and represents the language or dialect for which the preferences are defined. The reference set concept may be created in the extension, or may belong to a module on which the extension depends (e.g. the international core module).
referencedComponentId	SCTID	The referencedComponentId refers to the id of the description whose acceptability is being defined.
acceptabilityId	SCTID	The acceptabilityId indicates whether the description referenced by the referencedComponentId is |Acceptable| or |Preferred| for the given language or dialect.

Note some extension producers may need to create and populate *more than one* language reference set. For example, where different languages or dialects are used within a country, or where both clinician and patient friendly terms must be specified.

Example

In Figure 4.3.2.4.1-1 below, two rows from the 20581000087109 |Canada French language reference set| are shown. Note that these rows specify French Canadian language preferences for the concept |Herpes labialis|.

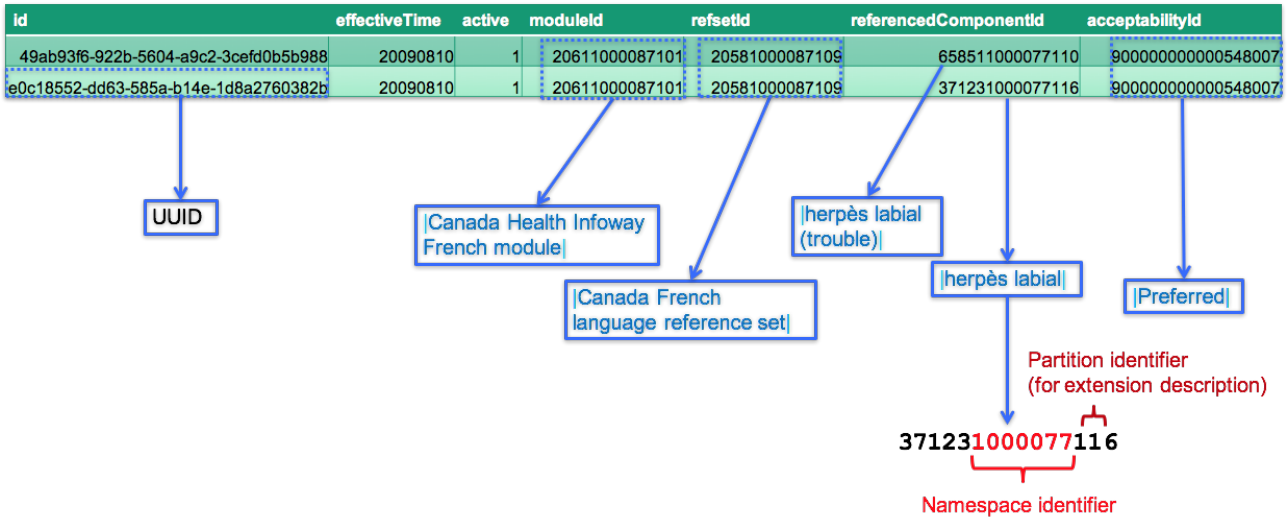


Figure 4.3.2.4.1-1: Rows from the |Canada French language reference set|

Please note the following:

- The id column shows a UUID that has been uniquely generated for each refset member.
- The moduleId indicates that these refset members belong to the |Canada Health Infoway French module|.
- The refsetId indicates that both members are part of the |Canada French language reference set|.
- The referencedComponentId indicates the description for which the acceptability is being specified.
- The acceptabilityId indicates that both of these descriptions are |Preferred| in this language.

4.3.2.4.2 Module Dependency Reference Set

Every extension must contain at least one module, which is used to organize the extension content. The dependencies that the extension modules have on other modules (including on the |SNOMED CT core module| and the |SNOMED CT model component module|) must be specified in the extension. This is done by adding new members to the |Module dependency reference set|. Each new versioned release of the extension requires a new version to be created for each extension member of the |Module dependency reference set|.

Attributes

As previously mentioned, all reference sets use six common attributes. In addition to these, a |Module dependency reference set| also includes a sourceEffectiveTime and targetEffectiveTime. The table below explains how each of these attributes is populated in a |Module dependency reference set|.

Table 4.3.2.4.2-1: Summary of module dependency reference set attributes

Attribute	Data Type	Use in a Module Dependency Reference Set

id	UUID	Can be generated using a standard UUID generator.
effectiveTime	Time	Specifies the date on which this version of the reference set member was released.
active	Boolean	Indicates whether or not the reference set member is active at the given effectiveTime. This value is set to "1" for new reference set members.
moduleId	SCTID	Identifies the module to which this reference set member belongs to. In the |Module dependency reference set|, this also represents the moduleId of the dependent module (which is the **source** of the dependency). The module dependencies are therefore always specified within the dependent module. This ensures that anyone who has access to an extension module can access the relevant dependencies.
refsetId	SCTID	In a |Module dependency reference set|, this always refers to the concept |Module dependency reference set|. All module dependencies are included in this common reference set.
referencedComponentId	SCTID	The referencedComponentId specifies the module that is the target of the dependency. The value of this attribute is the id of the module on which the source module (referred to by the moduleId attribute) is dependent.
sourceEffectiveTime	Time	Specifies the version of the dependent module (referred to by the moduleId attribute) for which the dependency applies. The snapshot view of the dependent module at the given sourceEffectiveTime represents the state of the terminology in which the dependency applies.
targetEffectiveTime	Time	Specifies the version of the target module (referred to by the referencedComponentId attribute) on which the dependent module depends. The snapshot view of the target module at the given targetEffectiveTime represents the state of the terminology in which the dependency applies.

Example

In Table 2 below, three rows from the |Module dependency reference set| in the 20170301 US Edition of SNOMED CT is shown. Please note that the SCTIDs in the moduleId and referenceComponentId columns have been replaced with their preferred terms for improved readability.

Table 4.3.2.4.2-2: Module dependency reference set rows

id	effectiveTime	active	moduleId	refsetId	referencedComponentId	sourceEffectiveTime	targetEffectiveTime
1244116f-fdb5-5645-afcc-5281288409da	20170731	1	900000000000207008 |SNOMED CT core module|	900000000000534007	900000000000012004 |SNOMED CT model component module|	20170731	20170731
5e28836d-1647-5e4c-8345-181f8696f0c4	20170901	1	731000124108 |US National Library of Medicine maintained module|	900000000000534007	900000000000207008 |SNOMED CT core module|	20170901	20170731
c4f97804-9fb2-52a8-8772-987e6db9c372	20170901	1	731000124108 |US National Library of Medicine maintained module|	900000000000534007	900000000000012004 |SNOMED CT model component module|	20170901	20170731

Please note the following:

- The first row states that the 20170731 version of the **source module** |SNOMED CT core module| is dependant on the 20170731 version of the **target module** |SNOMED CT model component module|.
- The second row states that the 20170901 version of the **source module** |US National Library of Medicine maintained module| is dependant on the 20170731 version of the **target module** |SNOMED CT core module| .
- The third row states that the 20170901 version of the **source module** |US National Library of Medicine maintained module| is dependant on the 20170731 version of the **target module** |SNOMED CT model component module|. Note that this is an example of a transitive dependency that must be explicitly stated.

4.4 Editions

A SNOMED CT edition is a complete set of SNOMED CT components and reference set members that belong to an identified focus module plus all of the modules on which the focus module depends. Every SNOMED CT edition includes the contents of zero or more extension modules and the contents of the International Edition. The International Edition itself includes the contents of the |SNOMED CT core module| and the |SNOMED CT model component module|.

An edition consists of a set of modules (usually maintained by different organizations) that can be used as a standalone terminology solution within a healthcare environment. For this reason, many extension producers choose to publish their extension as an edition (which includes the content on which the extension depends), rather than as a separate extension. As explained in 5.6.1.2 Packaging and File Naming, however, use cases exist for both packaging options.

Figure 4.4-1 below illustrates the composition and dependencies of a local edition that is composed of a local extension module and the two international modules.

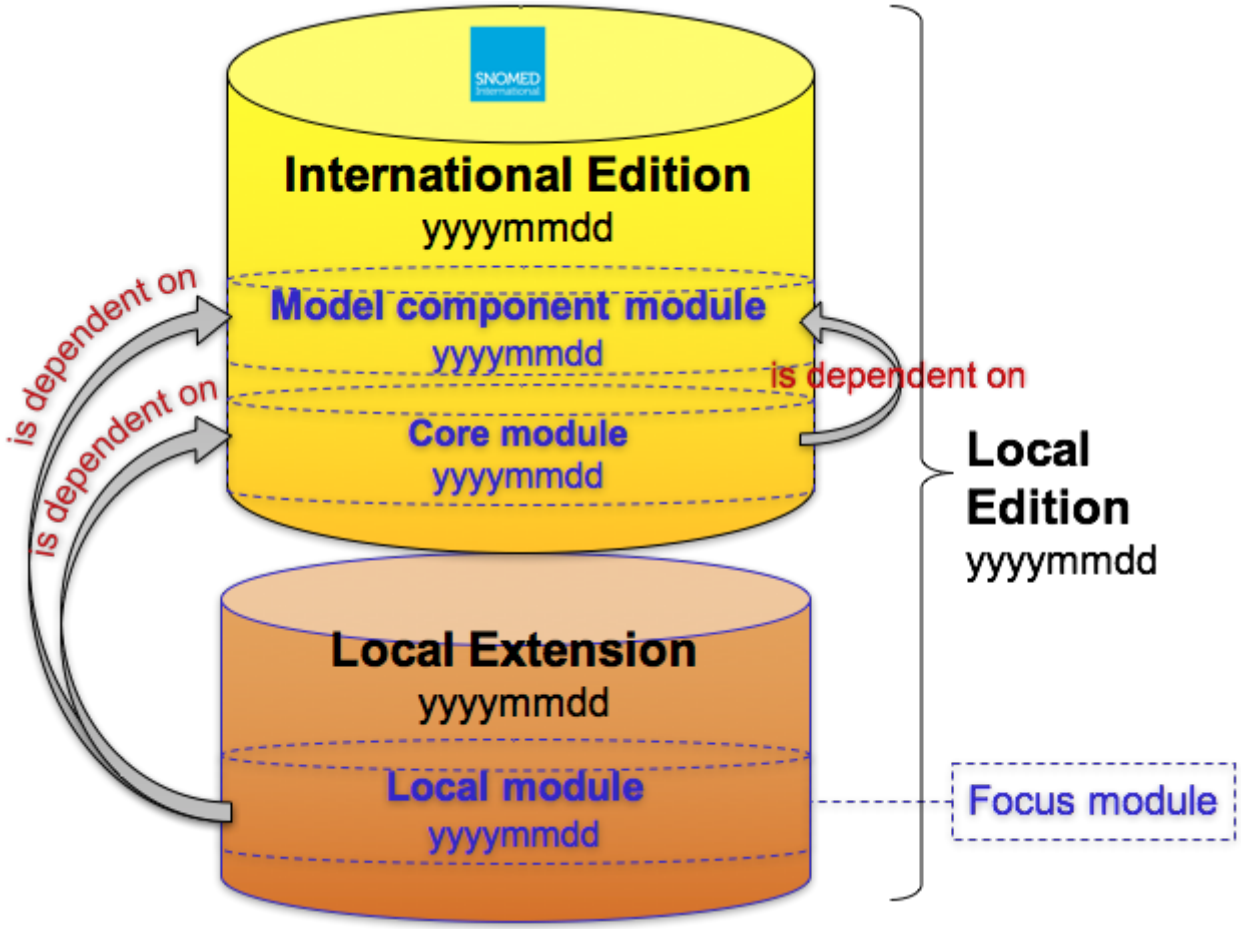


Figure 4.4-1: The composition of an edition is based on its module dependencies

The local edition consists of all components and reference set members that belong to the focus module and the two modules on which the focus module depends - |SNOMED CT core module| and the |SNOMED CT model component module|. As discussed previously, the version specific dependencies must be stated in the Module Dependency Reference Set.

Versioned Edition

There are many use cases that require a date specific version of an edition, including specifying the substrate of a SNOMED CT query, and specifying the version of SNOMED CT used to code a specific data element in a health record. A versioned edition includes the contents of the specified version of the focus module, plus the contents of all versioned modules on which the versioned focus module depends (as specified in the |Module dependency reference set|). The version of an edition is based on the date on which the edition was released. Many extension providers release their extensions as a versioned edition, using regular and predictable release cycles. For example, the International Edition is released by SNOMED International twice a year, on 31st January and 31st July.

Versioned editions can be identified using the SNOMED CT URI Standard using the following format:

http://snomed.info/sct/{sctid}/version/{timestamp}

where {sctid} represents the id of edition's focus module, and {timestamp} represents the release date of the edition. For more information, please refer to 2.1 URIs for Editions and Versions.

The table below shows some examples of versioned editions of SNOMED CT, with their corresponding focus module, URI and set of included modules.

Table 4.4-1: URIs for versioned editions of SNOMED CT

Edition	Focus module	URI	Included modules
International Edition, 20170731	|SNOMED CT core module|	http://snomed.info/sct/900000000000207008/version/20170731	Metadata + Core
International Edition, 20170731 (with ICD-10 maps)	|SNOMED CT to ICD-10 rule-based mapping module|	http://snomed.info/sct/449080006/version/20170731	Metadata + Core + ICD-10 maps
US Edition, 20170901	|US National Library of Medicine maintained module|	http://snomed.info/sct/731000124108/version/20170901	Metadata + Core + NLM maintained module
US Edition, 20170901 (with ICD-10-CM maps)	|SNOMED CT to ICD-10-CM rule-based mapping module|	http://snomed.info/sct/5991000124107/version/20170901	Metadata + Core + NLM maintained module + ICD-10-CM maps

Please note that a versioned edition is a logical composition of modules, and does not necessarily correspond to a single release package. For more information on release packages, please refer to 4.5 Release Packages and 5.6.1.2 Packaging and File Naming.

Focus Module

As discussed above, an edition is defined based on a single focus module. This focus module must be the most dependent module, in that the focus module is dependent on all the other modules in the edition. Note that the |SNOMED CT model component module| is always the least dependent module in all SNOMED CT editions, as it has no dependencies to other modules.

If a situation arises in which an extension has two equally dependent modules, then a new focus module may need to be created, with dependencies defined to all other modules in the edition. In this case, the focus module may contain only the components and reference set members used to define the module concept itself and its dependencies. Figure 4.4-2 below illustrates a situation in which two (equally dependent) sibling modules exist - national module A and national module B - which are both dependent on the |SNOMED CT core module| and the |SNOMED CT model component module|, and are not dependent on each other. To form a national edition, which contains both national module A and national module B, a new focus module must be created that is dependent on all other modules in the edition. The concept id of the new focus module is then used in the URI of the national edition that contains both national module A and national module B.

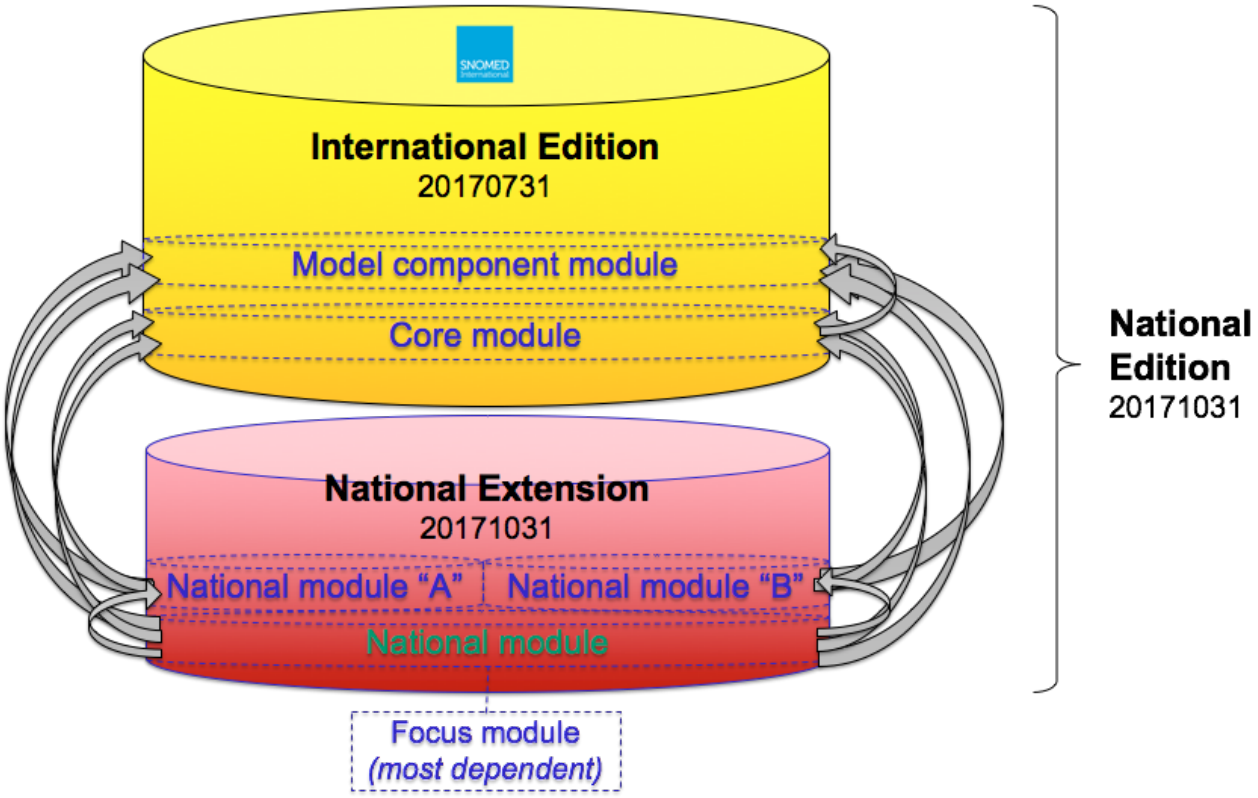


Figure 4.4-2: Creating a new focus module to represent a national edition

4.4.1 National Editions

National editions are published by a number of SNOMED International Members to customize the SNOMED CT International Edition to a specific country or territory. Table 4.4.1-1 below lists some examples of Members who have published national editions of SNOMED CT. Please note that the International Edition is translated into Spanish. However, this translation is performed by SNOMED International, and is therefore not distributed as a national edition.

Table 4.4.1-1: Examples of national SNOMED CT editions

	Australian edition		Netherlands edition		Canadian edition		UK edition
	Danish edition		Uruguay edition		US edition		Swedish edition

Table 4.4.1-2 below lists some examples of published SNOMED CT editions and their included modules, as specified in the Module Dependency Reference Set. Please note that a number of Members publish more than one edition. For example, in the UK both a national edition and clinical edition are published.

Table 4.4.1-2: Examples of national editions with their included modules

	Edition	Included modules
	International Edition, 20170731	|SNOMED CT model component module| + |SNOMED CT core module|
	United States Edition, 20170901	SNOMED CT International Edition: |SNOMED CT model component module| + |SNOMED CT core module|
		+ US Extension: |US National Library of Medicine maintained module| + |SNOMED CT to ICD-10-CM rule-based mapping module|
	Canadian Edition, 20171031	SNOMED CT International Edition: |SNOMED CT model component module| + |SNOMED CT core module|
		+ Canadian Extension: |Canada Health Infoway English module| + |Canada Health Infoway Reference Set Module| + |Canada Health Infoway French module|
	United Kingdom Clinical Edition, 20171001	SNOMED CT International Edition: |SNOMED CT model component module| + |SNOMED CT core module|
		+ UK SNOMED CT Clinical Extension: |SNOMED CT United Kingdom clinical extension module| + |SNOMED CT United Kingdom clinical extension reference set module|
	United Kingdom Edition, 20171108	SNOMED CT International Edition: |SNOMED CT model component module| + |SNOMED CT core module|
		+ UK SNOMED CT Clinical Extension: |SNOMED CT United Kingdom clinical extension module| + |SNOMED CT United Kingdom clinical extension reference set module|
		+ UK SNOMED CT Drug Extension: |SNOMED CT United Kingdom drug extension module| + |SNOMED CT United Kingdom drug extension reference set module| + |SNOMED CT United Kingdom Edition module| + |SNOMED CT United Kingdom Edition reference set module|

US Edition

Figure 4.4.1-1 below shows the four modules contained in the 20170901 US Edition, together with their module dependencies. Two of the modules are from the 20170901 US Extension, and the other two are from the 20170731 International Edition. Please note that the module dependencies (shown as grey arrows) are date specific, so each versioned source module is dependent on a versioned target module. The module names in this diagram have been shortened for brevity.

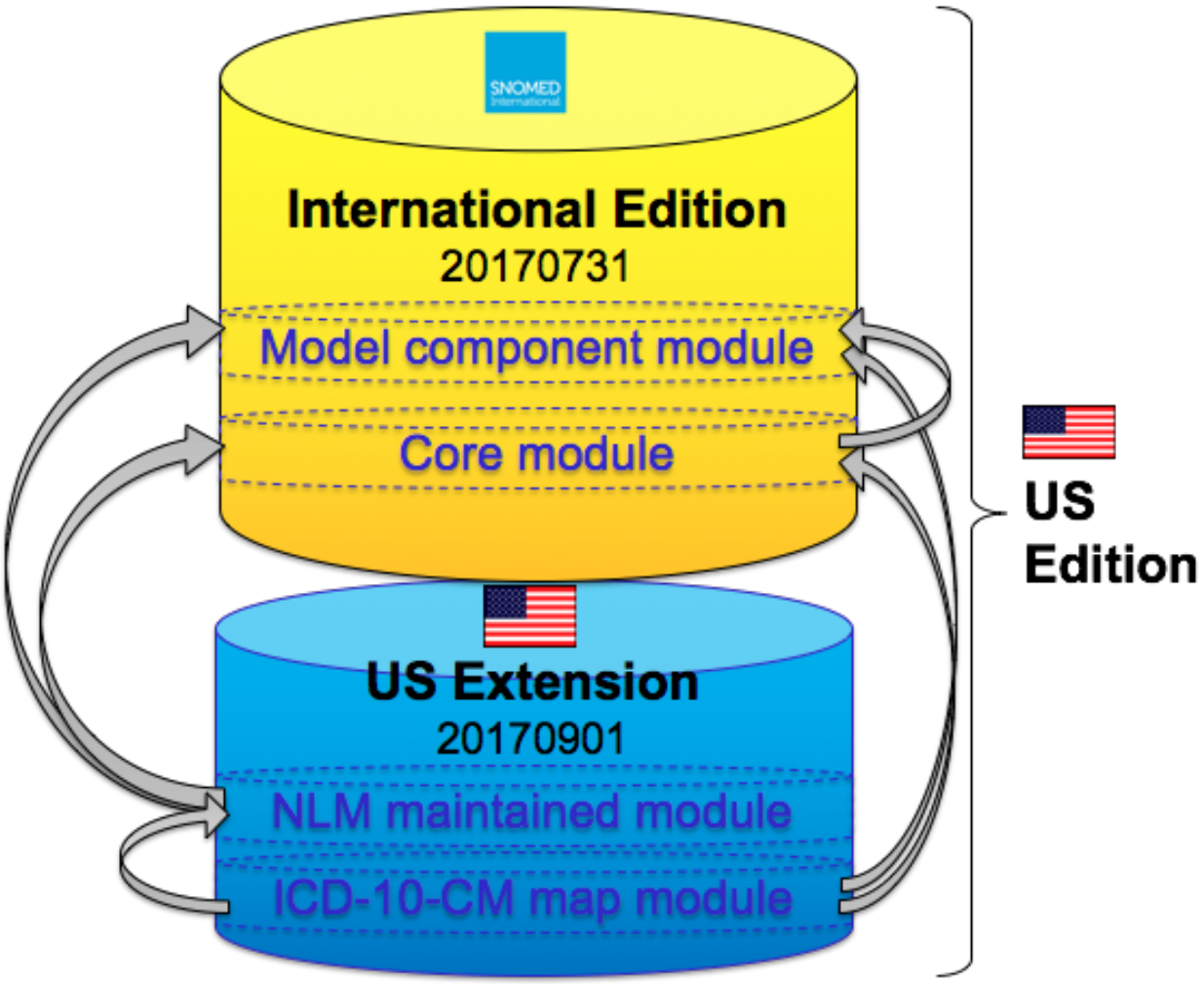


Figure 4.4.1-1: The modules in the US edition

Table 4.4.1-3 below lists the focus module and URI for the 20170901 US Edition.

Table 4.4.1-3: US Edition focus module and URI

Edition	Focus module	URI
US Edition, 20170901	SNOMED CT to ICD-10-CM rule-based mapping module	http://snomed.info/sct/5991000124107/version/20170901

Australian Edition

Figure 4.4.1-2 shows the five main modules in the 20171130 Australian Edition, together with their module dependencies. Two of the modules are from the 20170731 International Edition, and three are from the 20171130 Australian extension.

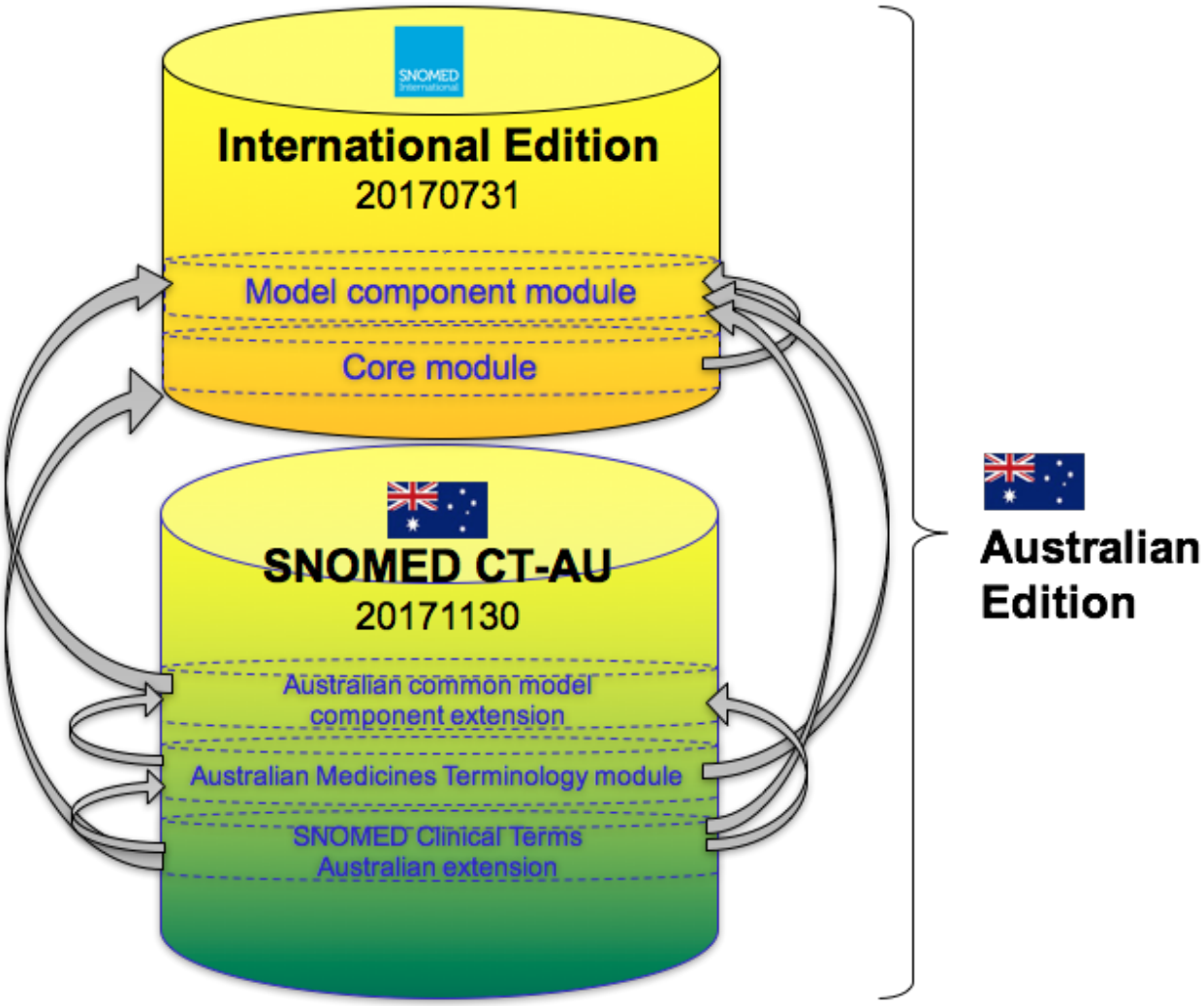


Figure 4.4.1-2: The modules in the Australian Edition

Table 4.4.1-4 below lists the focus module and URI for the 20171130 Australian Edition.

Table 4.4.1-4: Australian Edition focus module and URI

Edition	Focus module	URI
Australian Edition, 20171130	SNOMED Clinical Terms Australian extension	http://snomed.info/sct/32506021000036107/version/20171130

United Kingdom Editions

Figure 4.4.1-3 below shows the eight modules contained in either the 20171001 UK Clinical Edition or the 20171108 UK Edition, together with their dependencies. The 20171001 UK Clinical Edition contains 4 modules, including two from the International Edition and two from the UK Clinical Extension. The 20171108 UK Edition contains eight modules, including two from the International Edition, two from the UK Clinical Extension and four from the UK Drug Extension.

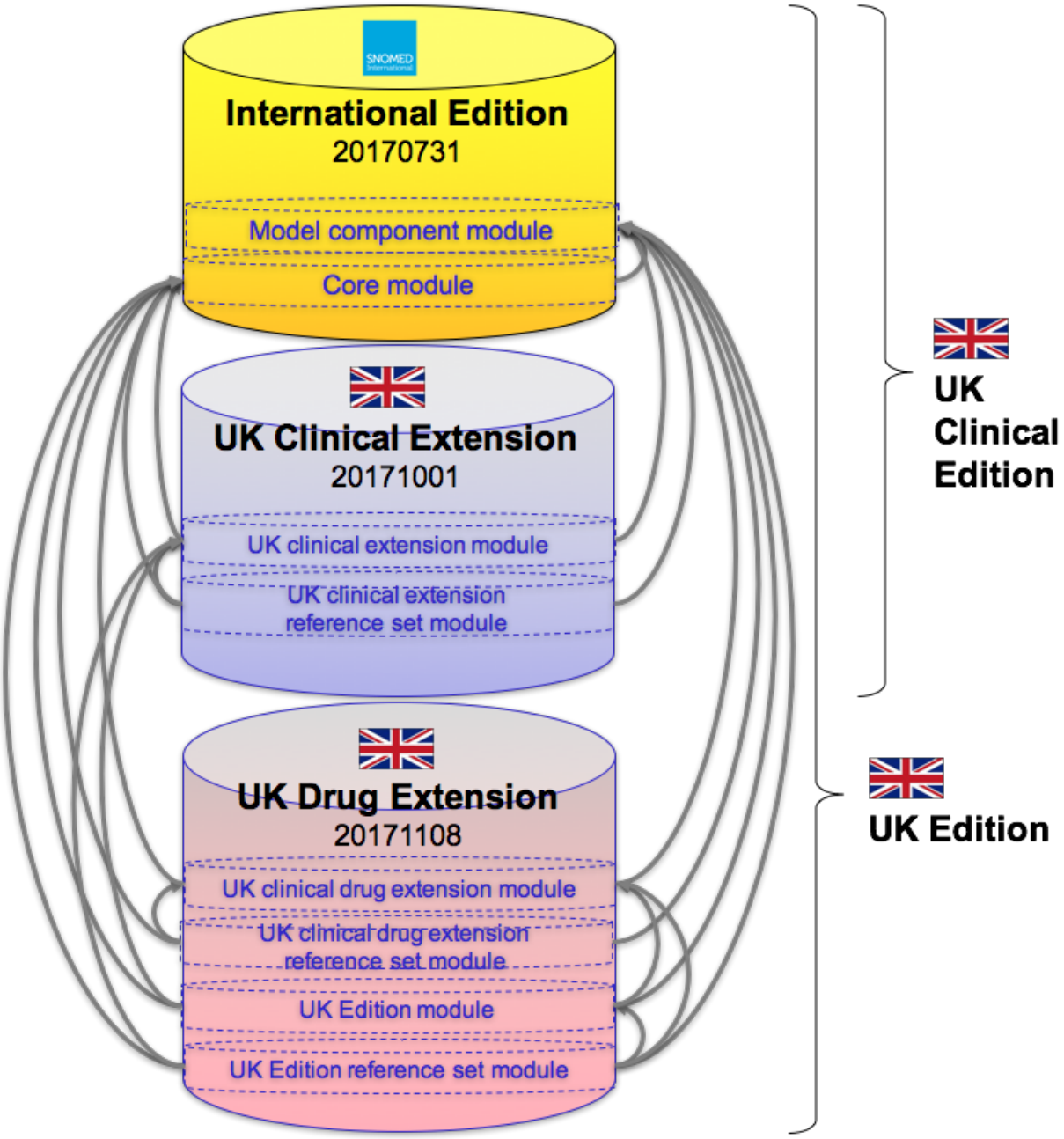


Figure 4.4.1-3: The modules of the two UK editions

Table 4.4.1-5 below lists the focus modules and URIs for the 20171001 UK Clincal Edition and 20171108 UK Edition.

Table 4.4.1-5: UK Edition focus modules and URIs

Edition	Focus module	URI
UK Clinical Edition, 20171001	SNOMED CT United Kingdom clinical extension reference set module	http://snomed.info/sct/999000021000000109/version/20171001
UK Edition, 20171108	SNOMED CT United Kingdom Edition reference set module	http://snomed.info/sct/999000031000000106/version/20171108

1. Note that although the |SNOMED CT to ICD-10 rule-based mapping module| is included in the International Edition release package, it is not considered part of the formal International Edition and extensions are not required to reference this module.
2. Note that the UK Clinical Edition contains the International Release and the UK SNOMED CT Clinical Extension of SNOMED CT only (not the UK Drug Extension). https://isd.digital.nhs.uk/trud3/user/guest/group/0/pack/26

4.5 Release Packages

A SNOMED CT release package is a set of SNOMED CT components, reference set members and other related items that are distributed in a set of release files as a single archived unit by SNOMED International, a National Release Center or another organization authorized to maintain and distribute a SNOMED CT extension. A release package may include one or more complete SNOMED CT Editions (in which all dependencies are resolved) or a supplementary extension module that depends on separately packaged modules. While a SNOMED CT edition is a logical construct consisting of the contents of a focus module and the modules on which it depends, release packages are the physical unit in which some or all of an edition is distributed. A release package, which is distributed at a given point in time, is referred to as a release package version or a versioned release package. For more information on release packages, please refer to 5.6.1.2 Packaging and File Naming.

For example, as illustrated in Figure 4.5-1 below, SNOMED International distributes the two modules of the International Edition plus the |SNOMED CT to ICD-10 rule-based mapping module| extension in a single release package, and distributes a range of other extensions (such as the 'International Classification of Nursing Practice (ICNP) to SNOMED CT' extension) as separate release packages. The National Library of Medicine (NLM) publishes the US Edition as a single release package, while NHS Digital (UK) currently publishes the UK Clinical Edition and the UK Drug Extension as separate release packages.

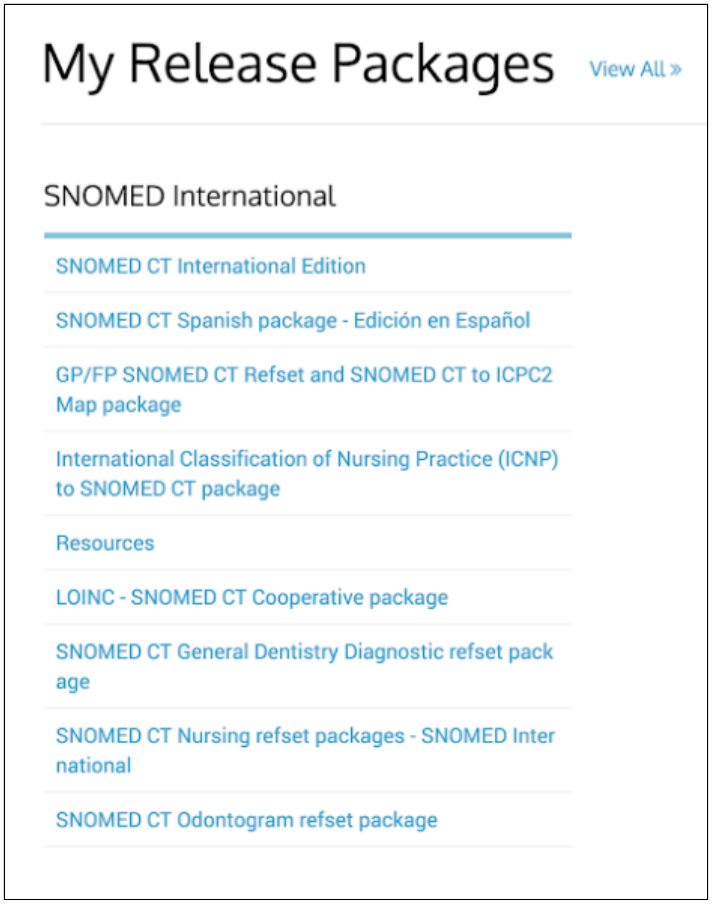


Figure 4.5-1: Release packages published by SNOMED International

5 Key Steps

While the lifecycle of each extension will vary due to differences in requirements and scope, three main phases can typically be identified - *Planning*, *Preparation* and *Production*.

1. The *planning phase* ensures that the requirements can be met by the extension design, resources and processes used.
2. The *preparation* phase ensures that the technical prerequisites are in place, including the namespace, module(s), module dependencies and appropriate tooling.
3. The *production phase* involves the development, distribution and maintenance of the extension.

The production phase can be further divided into several key steps, including
Assessing *Requests, Authoring*, *Review*, *Distribution*, and *Maintenance*.

1. *Assessing Requests* - Requests for new terminology products or content changes are submitted and processed to determine whether or not they will be accepted into the extension, submitted to a National Edition, submitted to the International Edition, or rejected.
2. *Authoring* - SNOMED CT components and reference set members are added, modified or inactivated according to the SNOMED CT editorial principles and policies. It should be noted that:
    - It is the responsibility of the extension producer to ensure that the quality and integrity of the extension is maintained, and that all content changes are made in a module that is owned by the *terminology producer* themselves.
    - All changes made to SNOMED CT components or reference set members should be represented as new versions and assigned to a module owned by the *terminology producer*. No changes are permitted to content of the International Release. Any modifications resulting in changes to the classification of international content must be accompanied by a disclaimer notifying users of the differences between the extension edition and the International Edition.
    - Any substantive improvements or corrections to the content in the International Edition that is made in an extension should be submitted to SNOMED International in a timely fashion to improve the quality of the International Edition for all users.
3. *Review and Validation* - SNOMED CT extension content should be validated using automated tests both at the time of authoring and before a release is packaged for distribution. In addition, manual review of the terminology content is also vital to ensure that it meets the quality standards for usability and clinical safety which cannot be tested automatically .
4. *Distribution* - This involves classifying the extension content[1], packaging the release files, validating the distribution package, and making the release package available to *terminology consumers*.
5. *Maintenance* - A SNOMED CT extension must be maintained to respond to new change requests, and to ensure that the consistency and integrity with the International Edition (and other modules on which it depends) is maintained appropriately.

These key phases and steps in the lifecycle of an extension are illustrated below in Figure 5-2. Further details about each of these key phases and steps are provided in the subsequent sections, together with the principles that should be followed to ensure the quality and integrity of a SNOMED CT extension.

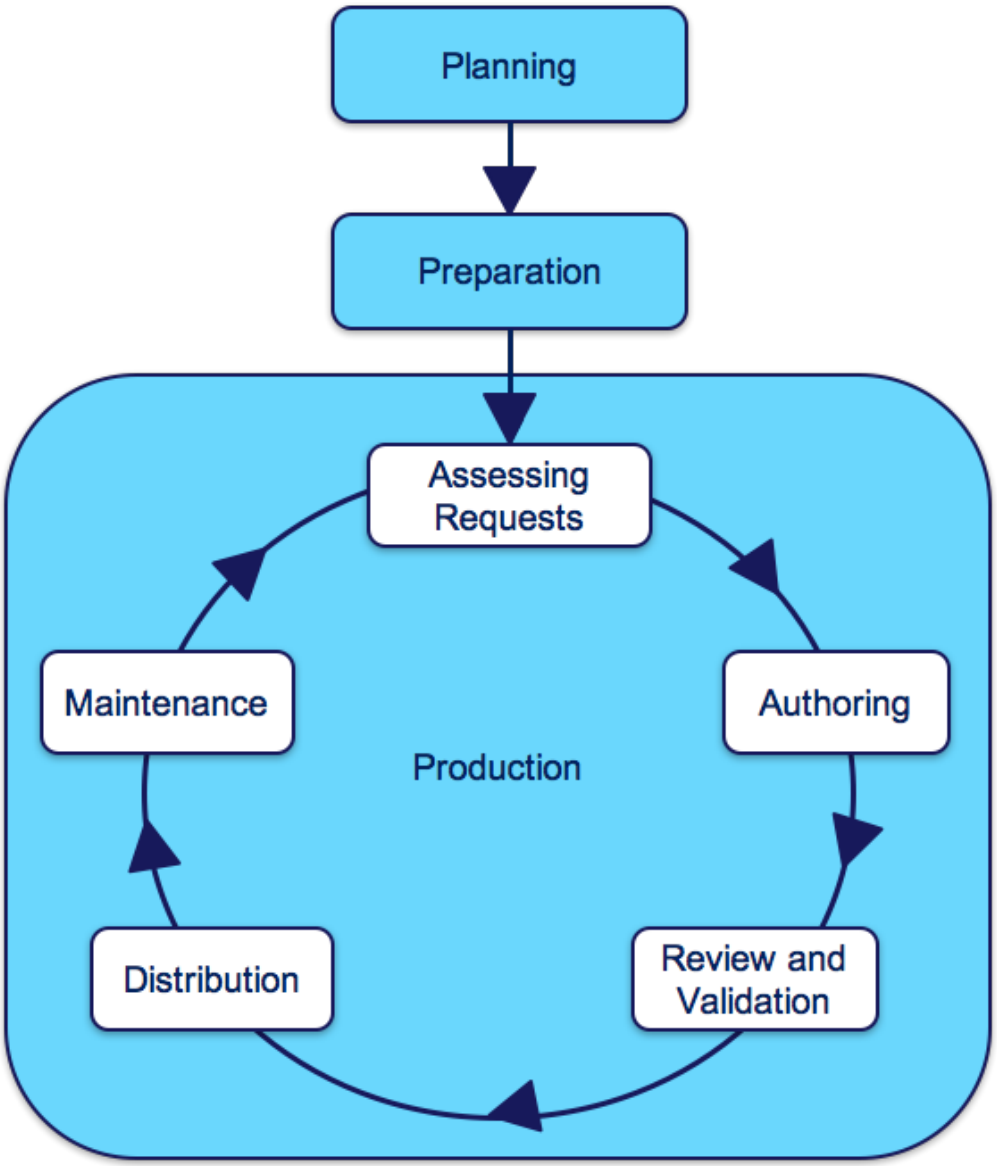


Figure 5-1: Extension Lifecycle Overview

In practice, classification of an extension module requires it to be combined with the all the modules on which it depends, including the International Edition modules.

5.1 Planning

Before starting the production of an extension, thorough planning is required to facilitate its successful development and subsequent use. This phase should include careful long-term and short-term planning, including consideration of the following topics:

- Scope and design of the extension
- Practical aspects related to extension development
- Ongoing operations and maintenance

The planning phase is about ensuring a balance between the scope and magnitude of the extension and the available resources, and about selecting appropriate methods and processes for the development, implementation

and maintenance of the extension. It is important to be as thorough as possible during this phase to avoid unforeseen issues and to mitigate potential issues in subsequent phases of the extension lifecycle.

Planning Topic		Key Questions to Address	Resources
Scope and design	Requirements	• What is the overall purpose of the extension? • Who are the main users of the extension?	Purpose of Extensions Chapter 4 - Requirements
	Content	• What is the scope of the extension? • What type of content will be included in the extension? - e.g. what components and reference sets should be developed and what types of files should be produced?	Chapter 3 - Scope and boundaries 4.3.1 Components
	Prerequisites and dependencies	• Is the extension dependent on other editions, such as the International edition, and if so which editions? • Are there other dependencies that influence the development of the extension? For example: • Is the extension dependent on other information artifacts, such as other terminologies or classifications, or specific information model standards?	4.2.2 Module Dependencies
Practical aspects of development	Resources	• What resources are required for the development and maintenance of the extension? *Note: The resources available should be determined (time, money, people) to ensure a proper balance between the resources available and the tasks involved during each phase of extension development, implementation and maintenance.*	

	Human resources	• What roles and competencies are required? For example:
		  • Clinical experts
		  • Terminology experts
		  • Technical and release experts	

	Development approach	• What methods will be used for authoring and reviewing the content in the extension and assuring the required level of quality?	5.5 Review and Validation
	Tools	• What tools will be used to support extension development, distribution and maintenance processes? *Note: A suite of tools may be required to support the creation and management of an extension. The functional requirements of these tools depend on the extension's scope, content and a range of other factors.*	5.2.1 Tooling Requirements

Planning Topic		Key Questions to Address	Resources

	Timelines	• When and how often will the extension be released?
		• What is the extension release cycle?
		  • What are the key tasks for each step?
		  • Who is responsible for each task during the release cycle?
		• What are the key deadlines in the release schedule?
		Note: It is important to consider both short-term and long-term requirements
			5.7 Maintenance

	Quality assurance	• What level of quality assurance (QA) is required, who will perform the QA, and when? • What review methods will be used?	5.5 Review and Validation
Operations	Distribution	• Will the extension be distributed as an extension or as an edition? • Does the release require classification to generate a combined inferred relationship file? • How will the standard file naming convention be applied? • How will the extension be packaged? • What tools or services will be used to distribute the extension?	5.6 Distribution
	Implementation guidance	• What type of guidance is required for extension consumers? • Who will develope the documentation and guidance material? (E.g. implementation guides, webinars, workshops)	5.6.3 Implementation Guidance

	Maintenance	• What methods and work processes are appropriate for the routine maintenance of the extension?
		• What types of changes will occur, and what tasks and services are required to manage these?
		• What is the release cycle for the extension and/or edition?
		  • Will the extension be updated with every version of the International Edition? (Note: This is recommended.)
		  • Will the extension be released following a fixed cycle, or will new versions be released on demand?
		Note: Please be aware that thorough validation and ongoing maintenance is required to ensure the long-term successful use of SNOMED CT.	

5.2 Preparation

Prior to creating the components or reference sets in an extension, some preliminary steps must be completed. These tasks include:

1. Obtaining a namespace identifier from SNOMED International
   - A namespace identifier is required to generate component identifiers (SCTIDs)
2. Acquiring the appropriate tools to support the creation, management, and distrbution of the extension
   - Which tools, functionalities and services are required will depend on the content of the extension
   - For more information please refer to Tooling Requirements

    - Note that these tools will be used to complete the steps listed below
3. Creating module(s)
    - At least one module concept is required
    - Components and reference sets which belong to the extension are assigned to the approriate module
4. Specifying module dependencies
    - These dependencies are specified using the Module Dependency Reference Set
    - This informs the consumers of the extension as to which additional modules are required to use/ process the content of the extension
        - Note that some of these modules will come from other editions (such as the International Edition) and possibly, other extensions

Figure 5.2-1 illustrates these key steps during the initial stage of extension creation.

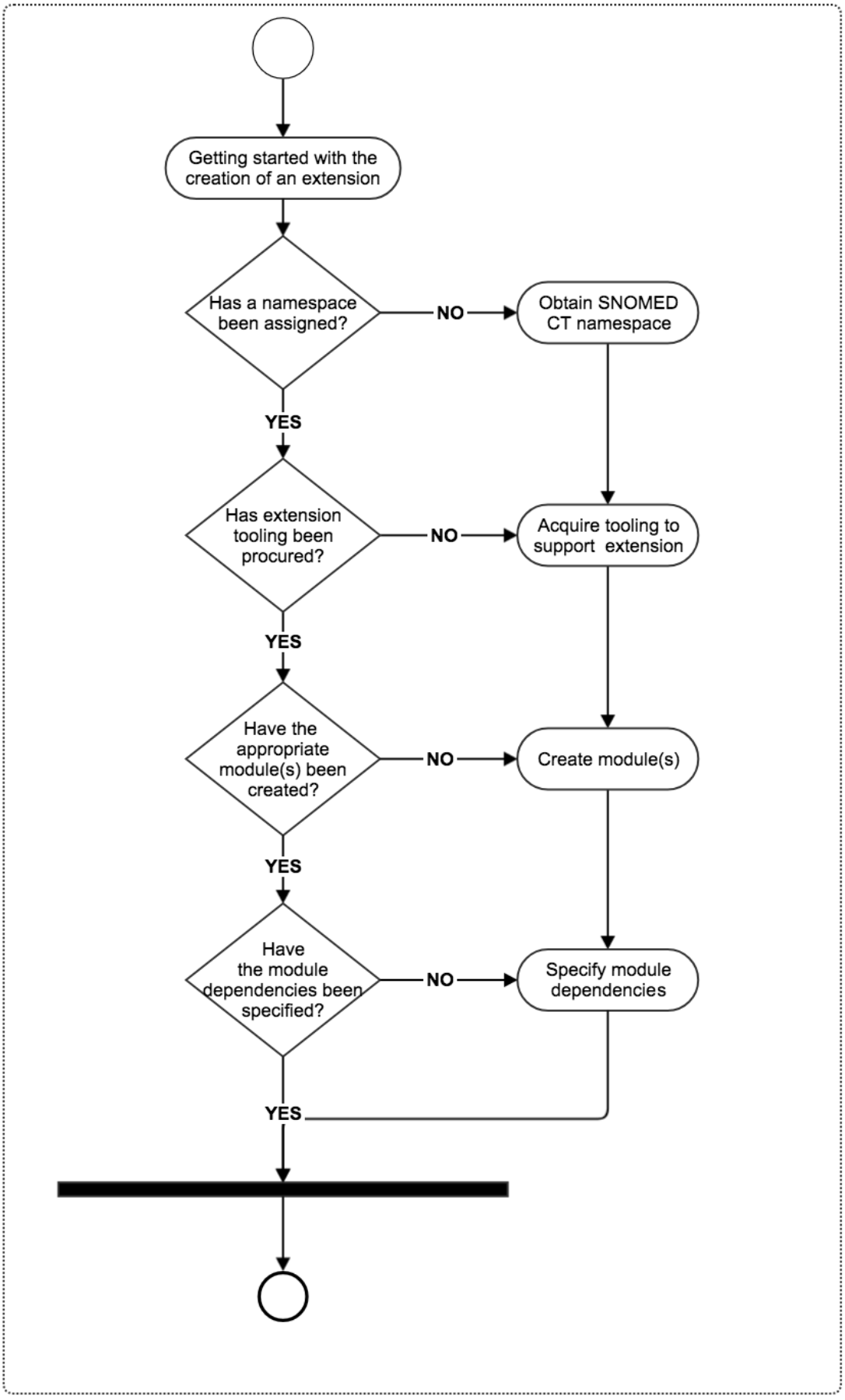


Figure 5.2-1: Prerequisites to creating an extension

5.2.1 Tooling Requirements

It is important that extension producers understand that some aspects of authoring and managing an extension are complex and will benefit from appropriate tooling.

Editions should not be handcrafted or managed using documents, files or simple spreadsheets, as these are likely to lead to technical errors and inconsistencies. Extension content needs to be reviewed in relation to its dependencies on the International Edition (and any other dependencies) which makes manual management and validation of extension content an unsustainable practice. Tools are required to maintain SNOMED CT content appropriately and to initiate automatic checks which can prevent errors.

As part of authoring and preparing for a release, all extension producers need tools capable of managing RF2 release file structures and performing necessary validation checks. Managing of RF2 files includes a mechanism for creating components and reference sets. There may be additional functionality required but this will vary depending on the content of the extension. For example, some extension producers producing new clinical concepts will need to be able to classify their extension to validate the created content and create the inferred relationships (distribution normal view) based on the stated relationships in the extension. Extensions which are merely used to distribute a translated version of SNOMED CT, or locally defined reference sets that only reference international release content will not need this classifying functionality as these extensions will only include primitive, metadata concepts. Please see the section on 5.6.1.1 Classifying an Edition for further information.

Table 5.2.1-1 summarizes the overall tooling considerations by category and services required. These are the tools required throughout the extension lifecycle including creation, management an distribution. Please note that this is not meant to be an exhaustive list, but instead summarize some of the key tools and services required.

Table 5.2.1-1: Extension tooling considerations

Category	Services/Mechanisms	Description
Authoring tools	Generate and manage SCTIDs	The ability to generate SCTIDs that are required for any component created within the extension. For more information, see generating identifiers.
	Generate and manage UUIDs	The ability to generate UUIDs that are required for any reference set member created within the extension. For more information, see generating identifiers.
	Author and maintain components	The ability to create concepts, descriptions and relationships in accordance with editorial principles. Note that varying levels of services and sophistication may be offered in tooling, but it is recommended to use tools which can demonstrate adherence to editorial principles. For example, a tool should ensure: • Compliance with the concept model • Component authoring rules such as concepts created with the mandatory subtype relationships and descriptions For more information, see authoring section.
	Author and maintain reference sets	The ability to create and maintain reference which comply with the reference set pattern as specified in the reference set descriptor and offer additional functionality such as the providing intensional definition of subset members For more information, see 6.3.2 Authoring Reference Sets.
Validation tools	Automatic validation of individual components	The ability to automatically validate that any terminology component present in the extension complies with the logical design of SNOMED CT, and that referential integrity is retained. For more information, see 5.5 Review and Validation.

Category	Services/Mechanisms	Description
	Automatic validation of the entire extension	The ability to automatically validate that the extension is ready for release. This includes ensuring the validity of release file structures, release file types, and referential integrity across all components and release files. For more information, see 5.5 Review and Validation.
Packaging and distribution tools	Classify extensions	The ability to classify SNOMED CT to create inferred relationships. For more information, see 5.6.1.1 Classifying an Edition
	Package extension edition	The ability to organize and package the edition according to the recommended release file structure. For more information, see 5.6.1.2 Packaging and File Naming.
	File or service-based distribution	The ability to distribute the edition, so extension consumers can access the terminology, either based on file distribution techniques or dedicated services to access specific terminology parts. For more information, see 5.6 Distribution.
Maintenance tools	Change request service	The ability to collect and process change requests, such as suggestions for new concepts, required synonyms for particular concepts, and new reference set members. These tools should also support the ability to forward internal change requests to SNOMED International when appropriate.
	Change identification mechanism	The ability to align the extension with the International Edition when new versions are published. This tooling includes the ability to identify changes to the International Edition which may affect the extension.

5.2.2 Generating Identifiers

Generating SNOMED CT Identifiers

The following guidance is provided for organizations which have been issued namespace identifiers to create content in their own extension:

- Organizations must only generate new SCTIDs for components using a namespace identifier assigned to them by SNOMED International
- Organizations must have a mechanism in place to ensure that each SCTID is assigned only once. Generally, a single authority that generates *item-identifiers* in a sequential fashion for each type of component will achieve this goal.
- *item-identifiers* should not be generated so as to have meaning. They should be regarded as meaningless numbers. (Recall that an item identifier precedes the namespace identifier in an SCTID.)

Generating UUIDs

Reference Set Members are identified with UUIDs instead of SCTIDs, so extension producers will also need the mechanism to generate UUIDs. UUID stands for Universally Unique Identifier, and each UUID is represented as a 128-bit number.

UUIDs are globally used to identify information in computer systems and they are for practical purposes unique, without depending on a central registration authority or coordination between the parties generating them. While the probability that a UUID will be duplicated is not zero, it is close enough to zero to be negligible.

Thus, anyone can create a UUID and use it to identify something with near certainty that the identifier does not duplicate one that has already been, or will be, created to identify something else. Information labeled with UUIDs by independent parties can, therefore, be later combined into a single database, or transmitted on the same channel, without needing to resolve conflicts between identifiers.

Several online platforms are available providing support for generating UUIDs, and for parsing their textual representation.

5.3 Assessing Requests

Once requirements have been identified and submitted as requests for additions or changes, the extension producer need to review and assess these requests. This entails an assessment of whether the requested content is relevant for inclusion in:

- The International Edition
- A National extension
- Another extension
- This extension (i.e. the one being managed by this extension producer)

Requests will typically be for addition of new components (concepts, descriptions or relationships) or changes to or inactivation of existing SNOMED CT components or reference sets. However, sometimes there are requests for a completely new reference set.

Important Considerations

As part of the assessment process to determine where the requested content belongs, these points should be taken into consideration:

1. The scope of various SNOMED CT Editions
2. Whether the request is permitted given the predefined principles around:
    a. Editorial rules and;
    b. Retaining the referential integrity of SNOMED CT

Scope

When a SNOMED CT Member, or Affiliate organization which maintains an extension, identifies a requirement for new content they should evaluate the request by considering the following questions:

- Does the requested content fall within the defined scope of the International Edition?
- Is the requested content likely to be required by, or of value to, more than one Member territory?

If the answer to both these questions is "yes", then organization should submit a request to SNOMED International for inclusion of the new content. When a request originates in an Affiliate organization, it should be submitted to a National Release Center who will then forward the request to SNOMED International. In general, this content should not be added to a National or local extension, as content that falls within the defined scope of the International Edition and is required by two or more Member territories. However, in cases where there is an urgent requirement, an extension producer may add the content to their own extension as an interim solution. Please refer to Interim Content Additions below.

Similarly, if the answer to one or both of these questions is "no", the content may be suitable for inclusion in a National Edition. In this case, the following questions should be considered:

- Does the requested content fall within the defined scope of a National Edition?
- Is the requested content likely to be required by, or of value to, more than one organization within the Member territory?

If the answer to both these questions is "yes", then organization should submit a request to the National Release Center for addition of the required content into the National Edition. If the answer to one or both of these questions is "no", it should be considered for inclusion in a local extension. In situation where the request is considered to be out of scope of SNOMED CT, it may be rejected. Table 5.3-1 provides additional detail on the various outcomes in these considerations.

Table 5.3-1: Considerations relating to appropriateness of content at various levels

Relevance of requested content	Description

Within the scope of the International Edition	If the content requested is within the scope of the International Edition, the organization should request that the content should be added to the International Edition. As SNOMED CT is a global clinical terminology, the scope of the International edition of SNOMED CT is constrained by a set of principles. For inclusion in the International edition, concepts must be • Useful in more than one country • Necessary for healthcare conformance and interoperability • Within the scope of a clinical terminology • Satisfy the criteria of being understandable, reproducible, useful If there is an urgent need for additional content, it may be added to an extension as an interim measure, pending addition to the International Edition. For more information, please refer to Interim content additions below.
Within the scope of other extension, (National, third-party, or local extension) but outside the scope of the International Edition.	The scope of a national edition may permit the addition of content which falls outside the scope of the International Edition but is required within a country, region or territory. If the content requested is within the scope of a national edition, a request should be submitted to the relevant National Release Center. An organization may also use an extension which was produced by another organization, such as a vendor, regional body or specialist group. In this case, requests for content which fall within the scope of these extensions should be submitted to the relevant extension producer. When there is an urgent need for additional content, it may be added to an extension as an interim measure, pending addition to a national edition or externally maintained extension. For more information, please refer to Interim content additions below.
Outside the scope of other extensions, but relevant to the organization requesting the content.	An organization which maintains an extension is responsible for defining the scope of that extension. The scope of an extension should not overlap with the scope of the International Edition or any other extension with modules on which the extension modules depend on. Note that urgent content additions may be permitted as an interim measure, even when the content is within the scope of another edition or extension. For more information. please refer to Interim content additions below. Requested content which falls within the defined scope of an extension may be added by the organization which maintains the extension. For additional information about adding local content to an extension, please refer to 5.4 Authoring.
Outside the scope of SNOMED CT	The request is rejected

Referential Integrity and Editorial Principles

When determining whether or not a request for additions or changes should be accepted for inclusion in any SNOMED CT edition, it is important to ensure that the addition or modification does not break the referential integrity of SNOMED CT. It is also important to assess whether the definition on new concepts has undesirable effects on the existing content. Finally, it should be ensured that every addition of or modification to SNOMED CT concepts conform to the rules specified by the concept model. Section 5.4 Authoring describes key principles which ensure that referential integrity is retained and that components and reference sets are created or modified in an extension in accordance with the editorial principles.

SNOMED International does not recommend that extension producers make changes which result in modifications to the International release as this can cause interoperability issues. Substantive errors, or improvements to content in the International release which are mitigated by content in an extension, should therefore be forwarded to SNOMED International in a timely fashion to improve the quality of the International release.

In circumstances where the International release has been impacted by modifications in a national or local extension, a disclaimer notifying users of the differences between the extension release and the International release must accompany the national extension distribution files.

⚠ Final disclaimer goes here: (proposal found here: Re: SNOMED International response to "Discussion paper - Allowance of Extensions to Modify Core Content" ver. 2.0)

Interim Content Additions

Before content is added to a national or International Edition, organizations may choose to add this content to a module they own as an interim measure. In these cases, the organization should indicate, as part of the request, that an interim addition has been made, but that the content should ultimately be promoted to the International Edition. When a request for content which is relevant at the national level is submitted to a NRC that maintains a National Edition, interim content from a local extension may be promoted to that National Edition.

For example:

- Local organizations in a country or region may add components to their local extension and then forward the request to their NRC
- An NRC may add components in their national extension and then forward a request to SNOMED International for inclusion in the International Edition

Note that, content added to an extension can later be promoted to another extension or to the International Edition. For more information, please refer to 5.7.1.1 Moving Components Between Extensions.

Change Request Reporting Structure

It is the responsibility of the extension producer to determine which requests are appropriate for their extension vs those which may be appropriate at a higher level. It is therefore important that extension producers are aware of the scope and the editorial principles for the International Edition of SNOMED CT and in some cases a national edition. This will ensure that requests for changes are forwarded to the relevant organization. Furthermore, it is important that extension producers to have an established and reliable process for evaluating, processing, and overall tracking of all change requests received. The process needs to have a routing mechanism to ensure that changes applicable to the extension are made in the extension, and external change requests can be forwarded to the responsible organization, such as SNOMED International.

Figure 5.3-1 illustrates the reporting structure for SNOMED International content requests. In general, only NRCs report to SNOMED International directly. Other extension producers will report to SNOMED International indirectly. However in somes cases, an extension producers may submit change requests directly to SNOMED International. For example, if the extension producer is an Affiliate, based in a non-Member territory.

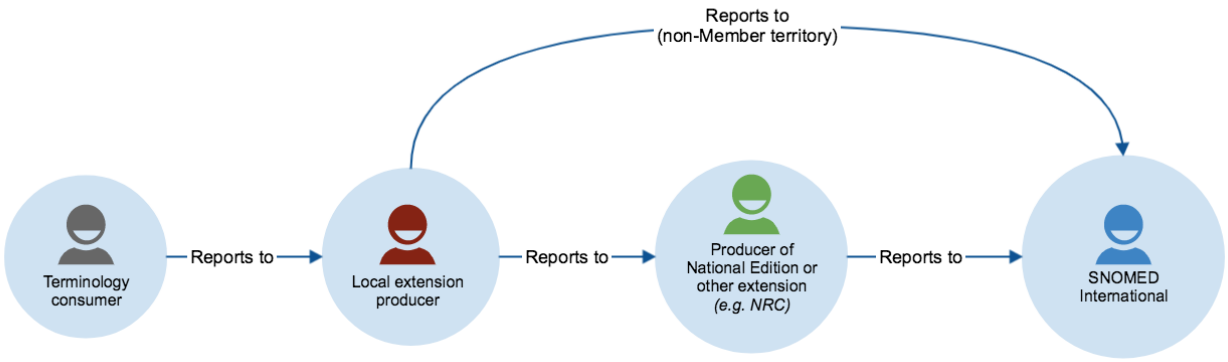


Figure 5.3-1: Change request reporting structure

National Release Centers are responsible for collecting requests for changes detected by Affiliates. Typically requests come from the users or consumers of the extension as they are most familiar with local requirements. Organizations within Member countries can therefore submit their requests for additions and changes to the National Release Center who will make an initial decision about the relevance of the content in the International Edition or the national edition of that country. The NRC may add content to their National Edition if

they consider the relevance to be national rather than international. The NRC forwards requests with (what they consider to have) international relevance to SNOMED International for consideration.

For more information about the processes in each country, please consult the relevant Member page.

SNOMED International provides the SNOMED CT Content Request Service to gather and process requests for additions and changes to the content of the SNOMED CT International Edition. This service is directly accessible by National Release Centers (NRC) in Member countries and recognized Terminology Authorities within organizations with whom SNOMED International is actively collaborating. Requests that meet inclusion criteria for the International Release are addressed by SNOMED International staff. Some of the reasons behind International content requests are described in Table 5.3-2.

Table 5.3-2: Reasons for requesting content to be added to the International Edition

Example use case	Description
Gaps in coverage of scope	Clinical meanings that fall within the scope of the International Edition that are not currently represented by concepts in the International Edition
Changes in clinical knowledge	Evolving clinical knowledge introduces clinical meanings that require the addition of new SNOMED CT concepts
Incomplete definitions	A concept in the International Edition that is missing one or more valid defining relationships
Missing synonyms	A concept in the International Edition that is missing one or more widely used synonymous terms

If a request is deemed to have high priority, it should result in action in the next release cycle. However, requests constituting a significant change which impacts other content may take longer to implement. If a request is declined, a reason and explanation are provided to the requester, who may choose to appeal the decision to the Head of Terminology. For more information about the CRS, please refer to the CRS User Guide.

5.3.1 Considerations in Assessing Requests

As part of the assessment process to determine where the requested content belongs, following points are important to take into consideration:

1. The scope of various SNOMED CT Editions
2. Whether the request is permitted given the predefined principles around:
    a. Editorial rules and;
    b. Retaining the referential integrity of SNOMED CT

Scope

When a SNOMED CT Member, or Affiliate organization which maintains an extension, identifies a requirement for new content they should evaluate the request by considering the following questions:

- Does the requested content fall within the defined scope of the International Edition?
- Is the requested content likely to be required by, or of value to, more than one Member territory?

If the answer to both these questions is "yes", the organization should submit a request to SNOMED International for the inclusion of the new content. When a request originates in an Affiliate organization, it should be submitted to a National Release Center who will then forward the request to SNOMED International. In general, this content should not be added to a National or local extension, as content that falls within the defined scope of the International Edition and is required by two or more Member territories. However, in cases where there is an urgent requirement, an extension producer may add the content to their own extension as an interim solution. Please refer to Interim Content Additions below.

Similarly, if the answer to one or both of these questions is "no", the content may be suitable for inclusion in a National Edition. In this case, the following questions should be considered:

- Does the requested content fall within the defined scope of a National Edition?

- Is the requested content likely to be required by, or of value to, more than one organization within the Member territory?

If the answer to both these questions is "yes", then the organization should submit a request to the National Release Center for the addition of the required content into the National Edition. If the answer to one or both of these questions is "no", it should be considered for inclusion in a local extension. In a situation where the request is considered to be out of the scope of SNOMED CT, it may be rejected. Table 5.3.1-1 provides additional detail on the various outcomes in these considerations.

Table 5.3.1-1: Considerations relating to appropriateness of content at various levels

Relevance of requested content	Description
Within the scope of the International Edition	If the content requested is within the scope of the International Edition, the organization should request that the content should be added to the International Edition. As SNOMED CT is a global clinical terminology, the scope of the International edition of SNOMED CT is constrained by a set of principles. For inclusion in the International edition, concepts must be • Useful in more than one country • Necessary for healthcare conformance and interoperability • Within the scope of a clinical terminology • Satisfy the criteria of being understandable, reproducible, useful If there is an urgent need for additional content, it may be added to an extension as an interim measure, pending addition to the International Edition. For more information, please refer to Interim content additions below.
Within the scope of another extension, (National, third-party, or local extension) but outside the scope of the International Edition.	The scope of a national edition may permit the addition of content which falls outside the scope of the International Edition but is required within a country, region or territory. If the content requested is within the scope of a national edition, a request should be submitted to the relevant National Release Center. An organization may also use an extension which was produced by another organization, such as a vendor, regional body or specialist group. In this case, requests for content which fall within the scope of these extensions should be submitted to the relevant extension producer. When there is an urgent need for additional content, it may be added to an extension as an interim measure, pending addition to a national edition or externally maintained extension. For more information, please refer to Interim content additions below.
Outside the scope of other extensions, but relevant to the organization requesting the content.	An organization which maintains an extension is responsible for defining the scope of that extension. The scope of an extension should not overlap with the scope of the International Edition or any other extension with modules on which the extension modules depend on. Note that urgent content additions may be permitted as an interim measure, even when the content is within the scope of another edition or extension. For more information. please refer to Interim content additions below. Requested content which falls within the defined scope of an extension may be added by the organization which maintains the extension. For additional information about adding local content to an extension, please refer to 5.4 Authoring.
Outside the scope of SNOMED CT	The request is rejected

Referential Integrity and Editorial Principles

When determining whether or not a request for additions or changes should be accepted for inclusion in any SNOMED CT edition, it is important to ensure that the addition or modification does not break the referential integrity of SNOMED CT. It is also important to assess whether the definition of new concepts has undesirable effects on the existing content. Finally, it should be ensured that every addition of or modification to SNOMED CT concepts conform to the rules specified by the concept model. Section 5.4 Authoring describes key principles which ensure that referential integrity is retained and that components and derivatives are created or modified in an extension in accordance with the editorial principles.

SNOMED International does not recommend that extension producers make changes which result in modifications to the International release as this can cause interoperability issues. Substantive errors, or improvements to content in the International release which are mitigated by the content in an extension, should, therefore, be forwarded to SNOMED International in a timely fashion to improve the quality of the International release.

In circumstances where the International release has been impacted by modifications in a national or local extension, a disclaimer notifying users of the differences between the extension release and the International release must accompany the national extension distribution files.

⚠ Final disclaimer goes here: (proposal found here: Re: SNOMED International response to "Discussion paper - Allowance of Extensions to Modify Core Content" ver. 2.0)

5.3.2 Constraints on Concept Requests

The SNOMED CT International Edition contains some concept hierarchies that have fundamental roles in supporting the design of SNOMED CT. Additions of concepts to these hierarchies may impact the consistency and integrity of SNOMED CT. The following pages set out guidance on the addition of extension concepts within specific hierarchies. In some cases, this guidance may prohibit extensions from adding concepts in a specific hierarchy or at a particular level of a hierarchy. In other cases, the advice is provided on the use case for making particular additions and notes are included on the implications and potential risks arising from such additions.

These constraints are expressed in tables in the following pages, which are arranged to follow the overall hierarchical structure of SNOMED CT.

5.3.2.1 Top Level Concepts and Clinical Concepts

Top Level Concepts

Hierarchy	138875005 | SNOMED CT Concept (SNOMED RT+CTV3) |
Purpose	The immediate subtype children of the root concept represent the most general types of SNOMED CT concepts.
Concept Addition Rules	It is strongly recommended that new top level concepts (i.e. direct subtype children of the root concept) are *not added* in an extension.
Additional Notes	

Top Level Clinical Hierarchies

Hierarchies	All direct subtype children of 138875005 | SNOMED CT Concept (SNOMED RT+CTV3) | except 900000000000441003 | SNOMED CT Model Component (metadata) | .
Purpose	These concepts represent a distinct type of clinical concept
Concept Addition Rules	Subtype descendants of the top level clinical concepts may be added in an extension.
Additional Notes	Concepts added in these hierarchies should conform with: • Concept model rules applicable to the relevant concept domain • Quality standards applicable to all SNOMED CT clinical content. These concepts should be classified to generate a set of inferred relationships in distribution normal form, for inclusion in the extension distribution files.

Model Metadata Concept Hierarchies

Core Metadata Concept

Hierarchy	900000000000441003 | SNOMED CT Model Component |
Purpose	This concept is the supertype for all SNOMED CT metadata.
Concept Addition Rules	No new subtype children may be added in an extension. Please refer to the following pages for guidance on the addition of subtype descendants.

All SNOMED CT metadata concepts are subtypes of 900000000000441003 |SNOMED CT Model Component (metadata)|and include concepts, descriptions and relationships that are used to describe or provide additional information about SNOMED CT content and reference sets.
Table 1 below lists the subtypes of the 900000000000441003 |SNOMED CT Model Component (metadata)|hierarchy, and describes the content within these subhierarchies.

Table 1: Subhierarchies of |SNOMED CT Model Component| and their intended purpose

Subhierarchy	Purpose	Link to More Information
Core metadata concept (core metadata concept)	Provides structural information required to support International Edition data. This supporting information includes sets of enumerated values that apply to attributes of concepts, descriptions and relationships.	5.3.2.2.1 Core Metadata Hierarchy
Foundation metadata concept (foundation metadata concept)	Provides supporting metadata and structural information for *Reference Sets*.	5.3.2.2.2 Foundation Metadata Concept Hierarchy
Linkage concept (linkage concept)	Links two or more concepts together to express compositional meanings. All *concept* codes that can be used as a *Relationship Type* are included under |Linkage concept|. The attributes that are approved for use are the *Concept Model Attributes*.	5.3.2.2.3 Linkage Concept Hierarchy
Namespace concept (namespace concept)	Contains *concepts* that represent an assigned *Extension namespace identifier*.	5.3.2.2.4 Namespaces

The following pages provide information about the subtypes of 900000000000441003 |SNOMED CT Model Component (metadata)|. Each page includes tables which describe the purpose of the content within one of the sub-hierarchies, with examples from the International Edition. The tables also state rules about the additions that are permitted in an extension to the respective subhierarchy. Where additions are permitted to the subheirarchy, notes are added on practical uses cases for addition, potential problems that may arise from these additions and any other relevant considerations.

5.3.2.2.1 Core Metadata Hierarchy

Core Metadata Concept

Hierarchy	900000000000442005 | Core metadata concept |
Purpose	This concept is the supertype for metadata used directly by SNOMED CT components.

Concept Addition Rules	No new subtype children may be added in an extension. See tables below for concept addition rules for each of the subtypes of 900000000000442005 | Core metadata concept | .

Characteristic Type

Hierarchy	900000000000449001 | Characteristic type |
Purpose	This subhierarchy provides the values for Relationship.characteristicTypeId.
Examples	• 900000000000227009 | Additional relationship | • 900000000000006009 | Defining relationship | • 900000000000225001 | Qualifying relationship |
Concept Addition Rules	No new subtype descendants may be added in an extension.

Definition Status

Hierarchy	900000000000444006 | Definition status |
Purpose	This subhierarchy provides the v alues for Concept.definitionStatusId
Examples	• 900000000000073002 | Defined | • 900000000000074008 | Primitive |
Concept Addition Rules	No new subtype descendants may be added in an extension.

Description Type

Hierarchy	900000000000446008 | Description type |
Purpose	This subhierarchy provides the v alues for Description.typeId
Examples	• 900000000000550004 | Definition | • 900000000000003001 | Fully specified name | • 900000000000013009 | Synonym |
Concept Addition Rules	Additional types permitted provided these are additive and do not replace the synonym or fully specified name values
Concept Addition Uses Cases	• Different lengths of text for different purposes • Different formats for text including HTML • Specialized usage of text for a specific purpose

Modifier

Hierarchy	900000000000450001 | Modifier |
Purpose	This subhierarchy provides the v alues for Relationship.modifierId
Examples	• 900000000000452009 | All | • 900000000000451002 | Some |
Concept Addition Rules	No new subtype descendants may be added in an extension.

Module

Hierarchy	900000000000443000 | Module |
Purpose	This subhierarchy provides the concepts used to identify modules
Examples	• 900000000000207008 | SNOMED CT core | • 900000000000012004 | SNOMED CT model component | • 449081005 | SNOMED CT Spanish edition module | • 449080006 | SNOMED CT to ICD-10 rule-based mapping module | • 449079008 | SNOMED CT to ICD-9CM equivalency mapping module |
Concept Addition Rules	The addition of modules that form part of the extension is required. These new module concepts must either be direct children of 900000000000443000 | Module | or descendants of a child of 900000000000443000 | Module | that is owned by the same extension provider. It is not permitted to add modules as subtypes of modules maintained by another extension provider. Each module concept added within an extension must have an identifier that contains the extension's namespace identifier.
Concept Addition Uses Cases	The addition of at least one module concept is essential for all extensions. More than one module concept may be added to enable content to be organized into separate modules.

5.3.2.2.2 Foundation Metadata Concept Hierarchy

Foundation Metadata Concept

Hierarchy	900000000000454005 | Foundation metadata concept |
Purpose	This concept is a supertype for metadata used to support SNOMED CT reference sets.
Concept Addition Rules	No new subtype children may be added in an extension. See tables below for concept addition rules for each of the subtypes of 900000000000454005 | Foundation metadata concept | .

Reference Sets

Hierarchy	900000000000455006 | Reference set |
Purpose	The subtype children represent reference set types. Subtype descendents represent either reference sets of the type specified by their supertype parent or groups of reference sets of a type specified by their supertype parent.
Examples	Reference set types • 609331003 | Extended map type reference set | • 447562003 | ICD-10 complex map reference set | • 900000000000521006 | Association type | • 900000000000522004 | Historical association | *(This is a grouper concept not an actual reference set)* • 900000000000526001 | REPLACED BY association reference set | • 900000000000527005 | SAME AS association reference set | • 900000000000480006 | Attribute value type |
Concept Addition Rules	New subtype children may be added in an extension, as long as they represent a new and distinct Reference Set Type (i.e. not an individual reference set). New subtype descendants may be added in an extension, as long as they represent a Reference Set of the type defined by its supertype ancestors.

Concept Addition Uses Cases	• To create a new reference set • To define a new reference set type • To define a group of related reference sets
Additional Notes	When a new reference set type is defined, an appropriate set of reference set descriptor rows must be created in the reference set descriptor reference set to specify the order and format of the reference set columns. Furthermore, the intended use and format of the new reference set type must also be documented. Appropriate documentation should also be provided to describe the specific use of each individual reference set. The addition of reference set descriptor rows for individual reference sets is also recommended. However, this is optional if there are no additional constraints over and above those specified for its reference set type.

Reference Set Attributes

Hierarchy	900000000000457003 | Reference set attribute |
Purpose	The subtype children represent named attributes used to represent columns in reference sets. Also includes two special subtypes: 900000000000459000 | Attribute type | and 900000000000491004 | Attribute value | which are separately documented.
Examples	• 900000000000511003 | Acceptability | • 900000000000549004 | Acceptable | • 900000000000548007 | Preferred | • 900000000000501005 | Map group | • 900000000000536009 | Source effective time |
Concept Addition Rules	Subtype children can be added in an extension to represent new column names for use in additional reference set type definitions. See separate notes on subtype descendance of the special subtypes 900000000000459000 | Attribute type | and 900000000000491004 | Attribute value | .

Reference Set Attribute Types

Hierarchy	900000000000459000 | Attribute type |
Purpose	These concepts represent data types associated with reference set attributes.
Examples	• 900000000000459000 | Attribute type (foundation metadata concept) | • 900000000000460005 | Component type (foundation metadata concept) | • 900000000000461009 | Concept type component | • 900000000000476001 | Integer (foundation metadata concept) | • 900000000000465000 | String (foundation metadata concept) | • 762678002 | OWL 2 language syntax (foundation metadata concept) | • 707000009 | SNOMED CT parsable string (foundation metadata concept) | • 900000000000466004 | Text (foundation metadata concept) | • 900000000000475002 | Time (foundation metadata concept) | • 900000000000469006 | Uniform resource locator (foundation metadata concept) | • 900000000000474003 | Universally Unique Identifier (foundation metadata concept) |
Concept Addition Rules	No new subtype children or subtype descendants may be added in an extension. If new subtypes are required these should be requested for addition to the International Edition.

Reference Set Attribute Values

Hierarchy	900000000000491004 | Attribute value |
Purpose	These concepts represent values that can be applied to specified columns in a reference set.

Concept Addition Rules	Additional subtype children may be added in an extension to represent values applicable to a newly added reference set or reference set type. Subtype descendants should **not** be added in an extension, except where they are descendants of a subtype child added to support a newly added reference set.
Potential Problems	Adding subtype descendant to pre-existing subtype children would have the effect of changing the permitted value set for a pre-existing reference set type and reference sets defined as having that type.

5.3.2.2.3 Linkage Concept Hierarchy

Linkage Concept

Hierarchy	106237007 | Linkage concept |
Purpose	This concept is the supertype parent or ancestor of 410662002 | Concept model attribute | , 416698001 | Link assertion | , etc.
Concept Addition Rules	No new subtype children may be added in an extension See tables below for concept addition rules for the descendants of existing subtype children.

Concept Model Attributes

Hierarchy	410662002 | Concept model attribute |
Purpose	These concepts provide the values for Relationship.typeId and represent the type of a defining relationship.
Examples	• 260686004 | Method | • 260870009 | Priority | • 363704007 | Procedure site |
Concept Addition Rules	Additional attributes may be added in an extension, where required to support extensions to the SNOMED CT concept model.
Concept Addition Uses Cases	Extending the SNOMED CT concept model to allow sufficient definitions to be applied to concepts that are not sufficiently defined in the International Edition.
Additional Notes	Any attributes created in extensions need to be carefully managed and documented. Extension attributes may also need to be revised or inactivated to align with future changes to the concept model in the International Edition.

Link Assertions

Hierarchy	416698001 | Link assertion |
Purpose	These concepts represent clinical assertions about relationships between instances of clinical statement in an EHR. For example, they may assert that a particular symptom is caused by something.
International Edition Examples	• 417151001 | Has explanation | • 416271009 | Has problem member | • 416586004 | Has problem name | • 416083004 | Has reason | • 417569004 | Has support | • 416872009 | Is etiology for | • 417318003 | Is manifestation of |
Concept Addition Rules	Additional subtype children may be added in an extension. However, it is recommended that proposed additions be submitted to the International Edition, unless they are highly specific.
Concept Addition Uses Cases	Requirements to support additional types of assertion in clinical records

5.3.2.2.4 Namespaces

Namespace Concept

Hierarchy	370136006 | Namespace concept |
Purpose	Each concept in this subhierarchy represents an allocated SNOMED CT extension namespace.
Examples	• 373872000 | Core Namespace | • 370137002 | Extension Namespace 1000000 | • 370138007 | Extension Namespace 1000001 | • 384597007 | Extension Namespace 1000002 | • 413335000 | Extension Namespace 1000003 |
Concept Addition Rules	No new subtype children or descendants may be added in an extension. Namespaces concepts can only be created by SNOMED International to represent allocated namespace identifiers.

Please see section 4 Logical Design for a further introduction to namespaces.

5.3.3 Interim Content Additions

Before content is added to a national or International Edition, organizations may choose to add this content to a module they own as an interim measure. In these cases, the organization should indicate, as part of the request, that an interim addition has been made, but that the content should ultimately be promoted to the International Edition. When a request for content which is relevant at the national level is submitted to a NRC that maintains a National Edition, interim content from a local extension may be promoted to that National Edition.

For example:

- Local organizations in a country or region may add components to their local extension and then forward the request to their NRC
- An NRC may add components in their national extension and then forward a request to SNOMED International for inclusion in the International Edition

Note that, content added to an extension can later be promoted to another extension or to the International Edition. For more information, please refer to 5.7.1.3 Updating Extension.

5.3.4 Change Request Reporting Structure

It is the responsibility of the extension producer to determine which requests are appropriate for their extension vs those which may be appropriate at a higher level. It is therefore important that extension producers are aware of the scope and the editorial principles for the International Edition of SNOMED CT and in some cases a national edition. This will ensure that requests for changes are forwarded to the relevant organization. Furthermore, it is important that extension producers have an established and reliable process for evaluating, processing, and overall tracking of all change requests received. The process needs to have a routing mechanism to ensure that changes applicable to the extension are made in the extension, and external change requests can be forwarded to the responsible organization, such as SNOMED International.

illustrates the reporting structure for SNOMED International content requests. In general, only NRCs report to SNOMED International directly. Other extension producers will report to SNOMED International indirectly. However, in some cases, an extension producers may submit change requests directly to SNOMED International. For example, if the extension producer is an Affiliate, based in a non-Member territory.

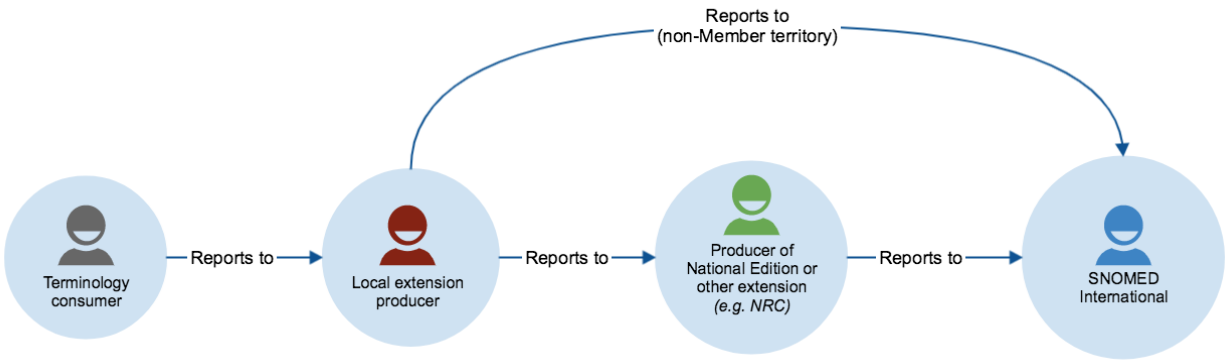


Figure 5.3.4-1: Change request reporting structure

National Release Centers are responsible for collecting requests for changes detected by Affiliates. Typically requests come from the users or consumers of the extension as they are most familiar with local requirements. Organizations within Member countries can, therefore, submit their requests for additions and changes to the National Release Center who will make an initial decision about the relevance of the content in the International Edition or the national edition of that country. The NRC may add content to their National Edition if they consider the relevance to be national rather than international. The NRC forwards requests with (what they consider to have) international relevance to SNOMED International for consideration.

For more information about the processes in each country, please consult the relevant Member page.

SNOMED International provides the SNOMED CT Content Request Service to gather and process requests for additions and changes to the content of the SNOMED CT International Edition. This service is directly accessible by National Release Centers (NRC) in Member countries and recognized Terminology Authorities within organizations with whom SNOMED International is actively collaborating. Requests that meet inclusion criteria for the International Release are addressed by SNOMED International staff. Some of the reasons behind International content requests are described in Table 5.3.4-1.

Table 5.3.4-1: Reasons for requesting content to be added to the International Edition

Example use case	Description
Gaps in coverage of scope	Clinical meanings that fall within the scope of the International Edition that are not currently represented by concepts in the International Edition
Changes in clinical knowledge	Evolving clinical knowledge introduces clinical meanings that require the addition of new SNOMED CT concepts
Incomplete definitions	A concept in the International Edition that is missing one or more valid defining relationships
Missing synonyms	A concept in the International Edition that is missing one or more widely used synonymous terms

If a request is deemed to have high priority, it should result in action in the next release cycle. However, requests constituting a significant change which impacts other content may take longer to implement. If a request is declined, a reason and explanation are provided to the requester, who may choose to appeal the decision to the Head of Terminology. For more information about the CRS, please refer to the CRS User Guide.

5.4 Authoring

Authoring is the process by which content is created in an extension in accordance with a set of authoring principles. These principles ensure the quality of content and referential integrity between content in the extension and content in the International Edition. It is the responsibility of an extension producer to author concepts,

descriptions and relationships in the extension while maintaining the aforementioned quality and referential integrity.

Terminology and extension producers need tools to assist with the authoring process, and many of the authoring principles described in the following sections can be automatically validated. It is therefore important to consider these principles when establishing requirements for an authoring tool. Furthermore, an extension producer should be aware of these principles and ensure that the content they develop in their extension does not directly modify any components or derivatives in a module belonging to another organization, such as the International Edition of SNOMED CT.

The following sections present key principles for authoring components and reference sets within an extension.

Please note that this section is not intended to teach how clinical content is authored. Clinical content is authored in accordance with editorial rules. For more information, please refer to the Editorial Guide.

5.4.1 General Authoring Principles

SNOMED CT components and reference set members are added, modified or inactivated according to the SNOMED CT editorial principles and policies.

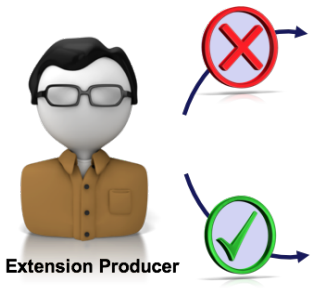
General principles that apply to all SNOMED CT extension content include:

- Organizations may add, inactivate or modify components and reference set members, which belong to modules owned by that organization.
    - It is the responsibility of the extension producer to ensure the quality and integrity of the extension is maintained, and that all content changes are made in a module that is owned by the extension producer themselves.
    - For more information about assigning modules in extensions, please refer to the section below on *Module Assignment*.
- No changes are permitted to content of the International Release, except for the addition of new versions of this content in a module owned by the extension producer.
    - Any modifications resulting in changes to the classification of international content must be accompanied by a disclaimer notifying users of the differences between the extension edition and the International Edition. Please note that modifications of this kind pose a risk to the comparability and interoperability of data captured using different SNOMED CT editions.
    - Any substantive improvements or corrections to the content in the International Edition that is made in an extension should be forwarded to SNOMED International in a timely fashion to improve the quality of the International Edition for all users.
    - Some reference sets, which belong to the International Edition (e.g. the |Module dependency reference set| and the |Concept inactivation indicator reference set|) were designed to be extended by adding new members in an extension module.
    - Other reference sets, which belong to the International Edition (e.g. |Great Britain English language reference set|) may be adapted in an extension module, following the principles described in 5.4.6 Authoring Reference Set Members

Module Assignment

Extension producers should assign all new extension content to a module which they own. Therefore every concept, description, relationship and reference set member, which is managed within an extension, should belong to a module created and owned by the extension producer. As illustrated in Figure 5.4.1-1, this is accomplished by setting the *moduleId* attribute in each component and reference set file to a module concept that was created by the extension producer in the given extension. Any dependencies between components or reference set members

in the extension module, and components or reference set members in other modules are specified in the module dependency reference set.



id	effectiveTime	active	moduleId	definitionStatusId
839999999105 |New concept|	20170731	1	900000000000207008 | SNOMED CT core module (core metadata concept) |	900000000000074008

id	effectiveTime	active	moduleId	definitionStatusId
839999999105 |New concept|	20170731	1	829999999108 | Extension module (foundation metadata concept) |	900000000000074008

Figure 5.4.1-1: Examples of correct and incorrect module assignment in an extension

5.4.1.1 Promotion and Demotion

SNOMED CT content may sometimes need to be moved from one module to another, to enable it to be published in a different edition. Moving SNOMED CT content **up** the module dependency chain (i.e. from a given module to a module on which it depends) is known as *content promotion*. This may be required to make the content accessable to a wider audience. Moving SNOMED CT content **down** the module dependency chain (i.e. from a given module to a module that is dependent on it) is known as *content demotion*. This may be required to make the content accessable to a narrower audience. In these situations, it is essential to be aware of the principles for promotion and demotion. These principles ensure the integrity and traceability of the content, in both the original module and the module to which it is moved.

Content Promotion

SNOMED CT components may be promoted from an extension module into a less dependent module to enable broader use. For example, an extension concept may be promoted to the International Edition to enable SNOMED CT users to share the concept. Please note that a component may only be promoted to a module on which the extension module depends (as specified in the module dependency reference set).

When promoting components from an extension into the International Edition, the donating organization (i.e., the owner of the extension) should submit a promotion request to SNOMED International with the details of all components to be promoted. The SNOMED International authoring team process the request and consider whether the content is acceptable for inclusion in the International Edition.

Once promoted, a new version of the component is available in the International Edition with a new *EffectiveTime*, an international *moduleId*, and the same SNOMED CT identifier as was used in the originating extension. From that point forward, SNOMED International becomes responsible for maintaining the promoted content. Future releases of the extension content alone, which are dependent on the new version of the International Edition (that contains the promoted content), will no longer include the promoted content in their Snapshot and Delta release files.[1] The promoted content will, however, continue to appear in the Full release files of the extension Edition with an older EffectiveTime to reflect the ownership history of this content within the extension. Please note that the SNOMED CT identifiers of all promoted content will remain unchanged, and will therefore continue to include the namespace identifier of the originating organization.

Please note that it is essential that the promoted content is not inactivated in the extension from which it was promoted, as this would make it appear to users of the extension that the content is inactive. It is also important that the meaning of the content remains the same after it has been promoted, to ensure that the component identifiers permanently represent the same clinical meaning.

The diagram below illustrates how extension content is promoted to the SNOMED CT International Edition.

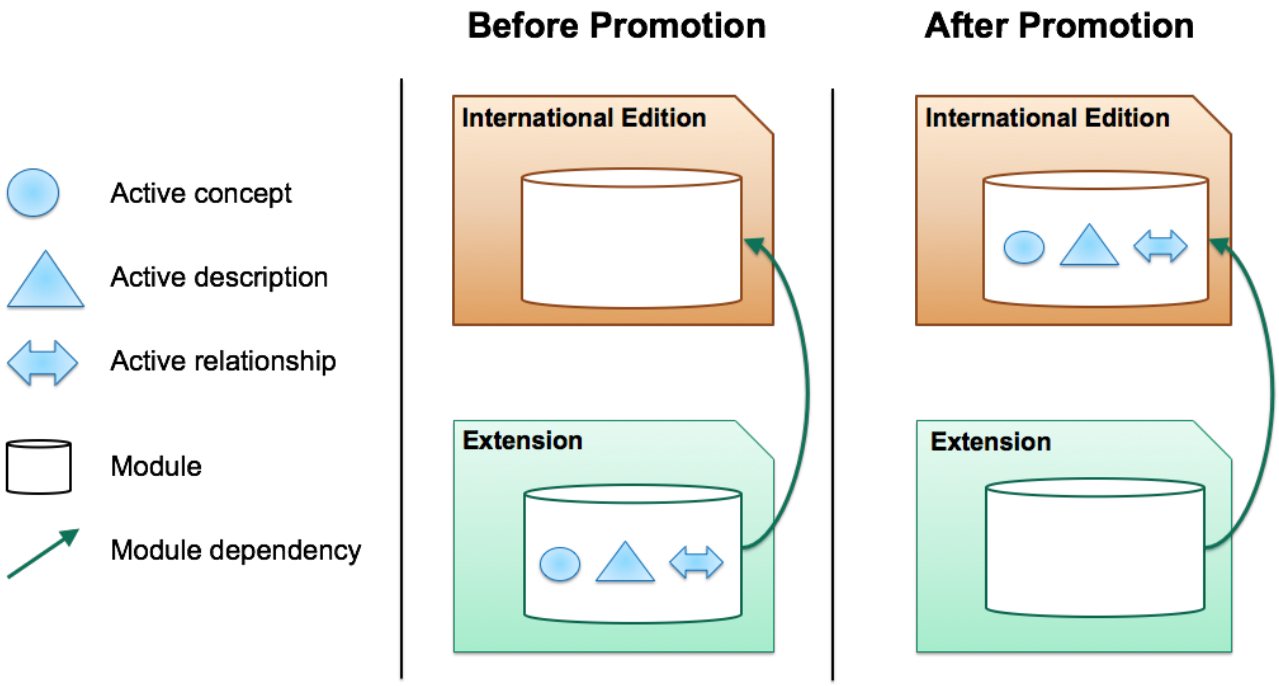


Figure 5.4.1.1-1: Promoting content to the International Edition

Content Demotion

In some circumstances it may be necessary to demote SNOMED CT components. Demotion means moving a component from a given module to a module that is dependent on it. For example, moving a concept (and its associated descriptions and stated relationships) from the International Edition to a National Extension.

Content demotion is generally not recommended, because there are several risks associated with the management of demoted content. Once demoted, a different version of the component will be seen in different editions, and this redundancy needs to be managed carefully. Consideration should, therefore, be given when making the decision to demote a component.

If content demotion is deemed necessary, then it is very important to use a safe approach. As shown in the diagram below, the recommended approach for demoting SNOMED CT content is to inactivate the component in the source module and activate the component in the destination module at a more recent effective time.

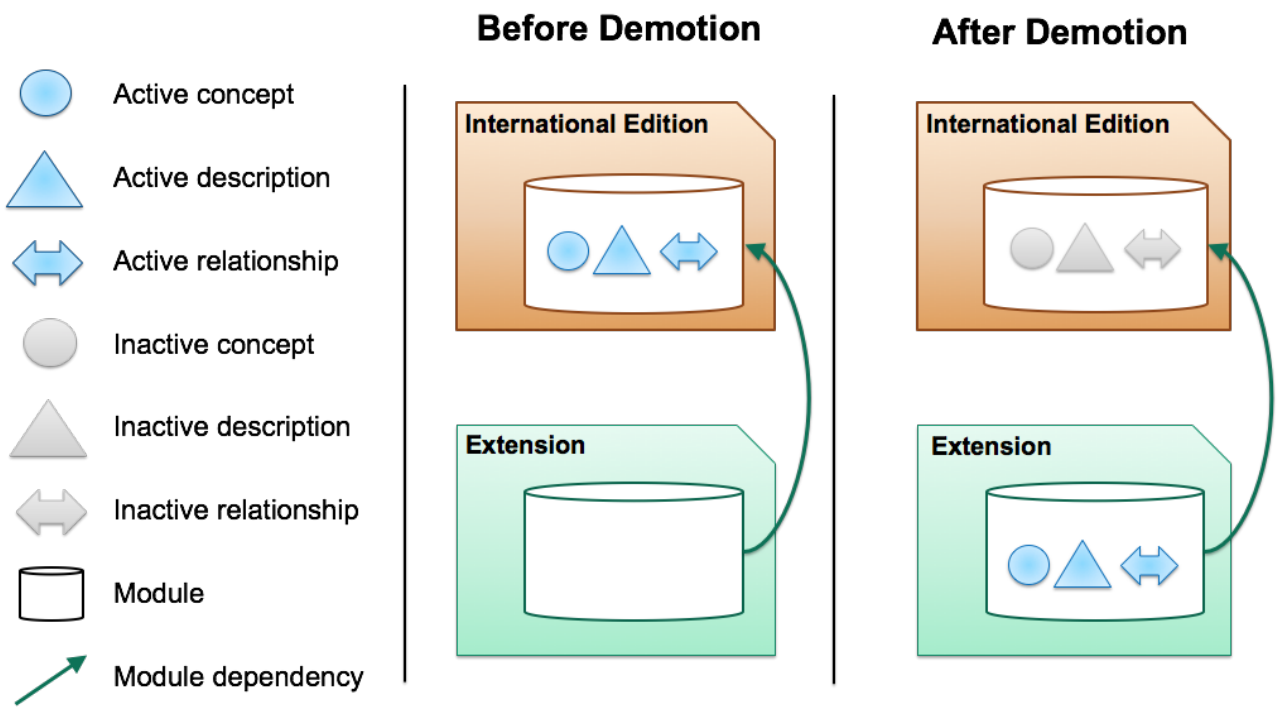


Figure 5.4.1.1-2: Demoting content from the International Edition to extension

Updates should also be made to the respective component inactivation reference sets to indicate that the component has been inactivated because |Component moved elsewhere|, and to the |MOVED TO association reference set| to indicate the module to which the component has been moved. For more information on inactivation and historical associated reference sets, see 3.2.6.3. Managing Component Inactivation.
The following diagram shows the RF2 file changes that are required to demote an example concept from the international edition to an extension module. Please note that it is essential that the effectiveTime of the active component in the destination module is more recent than the effectiveTime of the inactive component in the source module. This is done to ensure that the demoted component is seen as active in any edition that uses the extension module to which it was demoted.

International Edition - Concept file

id	effectiveTime	active	moduleId	definitionStatusId
839999999105 |Example concept|	**20180131**	0	900000000000207008 | SNOMED CT core module (core metadata concept) |	900000000000074008

Extension - Concept file

id	effectiveTime	active	moduleId	definitionStatusId
839999999105 |Example concept|	**20180331**	1	829999999108 | Extension module (foundation metadata concept) |	900000000000074008

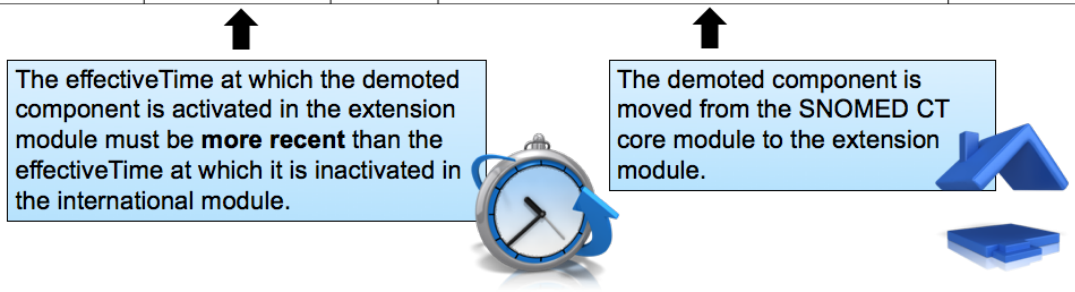


Figure 5.4.1.1-3: File changes required to demote a concept

Some approaches to content demotion that have previously been proposed are considered to be unsafe. The content demotion approaches that should **never** be used include:

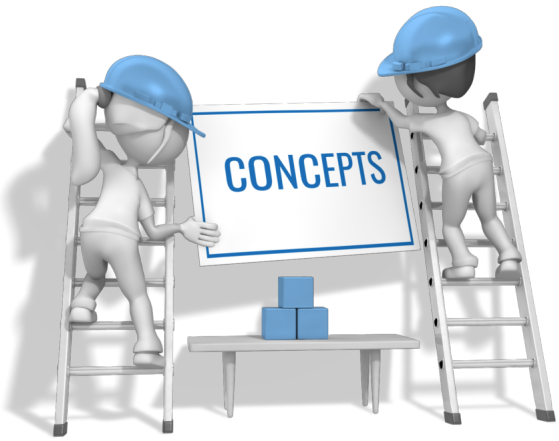
- Removing every instance of the component from the International Edition (or source edition)
  - This approach will delete the history of the component for users of the International Edition, who do not use the demoted content. The history of each SNOMED CT component should be traceable in the edition in which the component was created.
- Retaining the component active in the source module, and creating a new version of the component (i.e., with a more recent effectiveTime) in the less dependent module
  - This approach will make the component visible and accessible in the source edition, and users of the source edition would not be aware that the component was demoted to a less dependent module
  - This approach may lead to inconsistency between the version of the component in the International Edition and the version in the less dependent module.

Please note that if the extension is published as part of an Edition, in which the files include content from both the extension and International Edition, then the Snapshot and Delta release files will contain the promoted content as part of the international module.

5.4.2 Authoring Concepts

Authoring concepts in an extension may involve adding new concepts, inactivating concepts, or modifying existing concepts.

The sections that follow will examine the purpose, principles and process for each of these authoring tasks.



5.4.2.1 Add Concept in an Extension

Purpose

Concepts may be added to an extension to provide new clinical ideas which are not represented in the International Edition. The primary reason for adding concepts to an extension is to represent meanings which are outside the scope of the International Edition, yet within the scope of SNOMED CT. These might be concepts which have local relevance, within a single country, territory or organization. Concepts may also be added to an extension to represent metadata about the extension, for example module concepts, reference set concepts, or new attributes.

For a more detailed introduction on the purposes for adding concepts to an extension, please refer to Section 3 Purpose.

Principles

Overview

Key principles for authoring concepts in an extension include

- Concepts added to an extension must have a *moduleId* that includes the namespace identifier assigned to that extension producer by SNOMED International
- An extension concept must comply with the logical design of SNOMED CT components and maintain referential integrity
- Concepts added to an extension should conform to the policies stated in 5.3.2 Constraints on Concept Requests
- Use caution when authoring an intermediate concept in an extension, i.e. a concept which is a supertype of a concept in the International Edition

Essential Descriptions and Relationships

A concept created in an extension must conform to the logical design of SNOMED CT concepts. An overview of the logical model is provided in Figure 5.4.2.1-1. This means that every concept created in an extension must have:

- At least one |Is a| relationship
- At least one description of type |Fully specified name|
- At least one description of type |Synonym|

Additional Descriptions and Relationships

Concepts created in an extension may also have additional descriptions and defining relationships.

Additional defining relationships may include |Is a| relationships and attribute relationships. When specifying attribute relations for an extension concept it is important to ensure compliance with the SNOMED CT concept model. The concept model specifies the attributes that can be applied to particular types of concepts and it specifies the permitted values for each of these attributes. There are also additional rules on the cardinality and grouping of particular types of relationships. Please see the SNOMED CT Editorial Guide and the SNOMED CT Machine Readable Concept Model for details on the concept model rules.

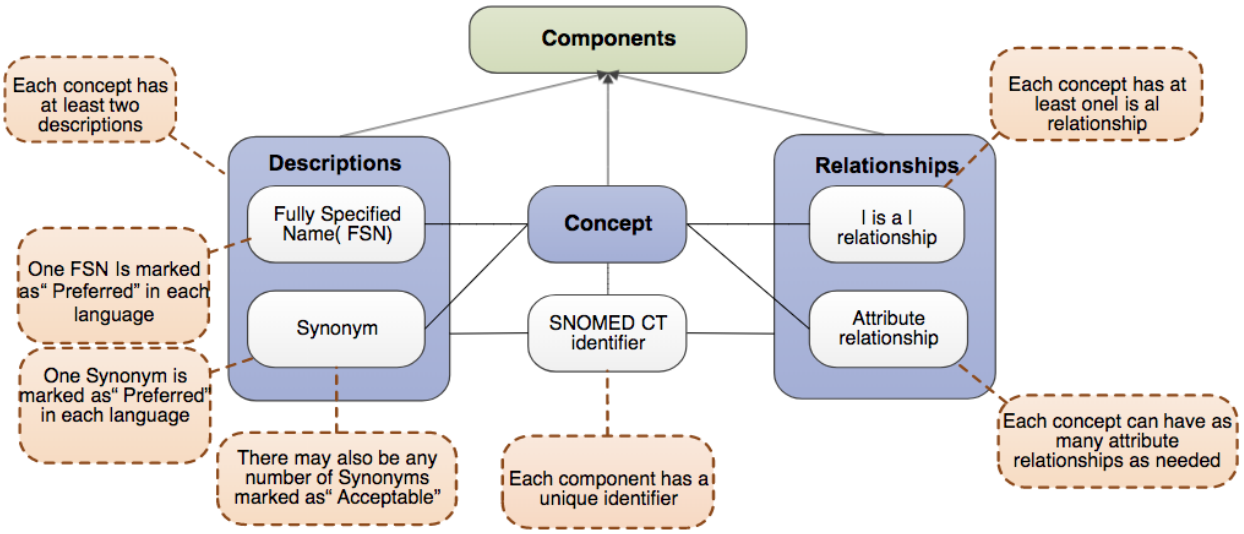


Figure 5.4.2.1-1: Logical Model Overview

Retaining Referential Integrity

As discussed in module dependencies, all extensions have modules which depend on the modules in the International Edition. Extensions may also have modules which depend on the modules from other extensions. Despite the dependencies between modules from various editions, all concepts in an extension must be a subtype of the root concept, 138875005 |SNOMED CT Concept|. This ensures that a concept in an extension can be queried and retrieved in a similar way to any concept in the International Edition. Put another way, queries which use subsumption should be able to identify a concept which belongs to a module in an extension because it is a subtype of a concept from the International Edition.

Extension concepts must be a subtype of the SNOMED CT root concept
The diagram illustrates how all concepts from Extension B are related to concepts in the International Edition. This is because the concept is either a direct child of a concept in the International Edition or because it is subsumed by concepts in the International Edition through subtype relationships defined by concepts in Extension A.

The referential integrity of a concept in an extension is retained by the addition of at least one |is a| relationship to a parent concept that is subsumed by |SNOMED CT Concept|. The parent may be a concept which is part of a module within the extension, another extension, or the International Edition.

Leaf and Intermediate Extension Concepts

When an extension producer creates a new concept and states one or more |is a| relationships to define that concept the immediate effect is to make the new concept a *leaf concept* in the SNOMED CT subtype hierarchy. That is to say the new concept at this point has no known subtypes.

There are several situations in which this new concept may acquire new subtypes:

1. Other concepts in the same extension module may be stated to be a subtypes of the new concept
2. The organization responsible for another module that depends on this extension module may, in the future, state that some concepts in their module are subtypes of this new concept
3. When a Description logic classifier is applied to the contents of the module (and the set of modules it depends upon), the defining relationships associated with a new fully defined concept may lead to additional subtype relationships being inferred. These inferred subtypes may be concepts in the extension module itself or in any module on which the extension module depends (including International Edition modules).
4. It is also possible (although caution is advised) that the creator of the new concept might state that the concept is a supertype of a concept in the International Edition or in another module on which the extension module depends.
5. it is also possible (although caution is advised) that an extension producer might add a defining relationship to a concept in the International Edition (or in another module on which the extension module depends), and this new defining relationship may lead to a variety of new relationships (including subtype relationships) being inferred by a Description logic classifier.

Situations 1 and 2 above both result in the subtypes of the new concept either belong to the extension module or a module that depends on the extension module. This does not create any exceptions.

In contrast, situations 3 and 4 can result in the new concept becoming an *intermediate concept* that sits between two concepts from the International Edition (or from another module on which the extension module depends). This means that the new concept is inferred to be a supertype of a concept from the International Edition (or another module that the extension module depends on). Since, all extension concepts are a subtype of a concept from the International Edition, the new concept represents an intermediate node in the inferred hierarchy between two International Edition concepts. This *intermediate concept* is only present in the editions that include the extension, so it does not directly alter the International Edition. However, it does create a situation in which the inferred view of the International Edition hierarchy seen by those using the extension differs from the view seen by other users of the International Edition.

Situation 3 does not impact the logical definition of the inferred subtypes in the International Edition since concepts will only be inferred as subtypes if they already share the logical definition of a fully-defined supertype. However, if (as in situation 4) an International Edition concept is stated to be a subtype of the new extension concept, the impact can be more profound. In this case, the logical inference is that all defining attribute relationships of the new concept must also apply to the subtype concepts. Finally, situation 5 directly changes the logical definition of a concept from a module published by another organization. This can have a profound effect not only on that concept itself, but also on subtypes of that concept and on the inferred subtype hierarchy of the International Edition as seen by users of the extension.

In summary, extension producers should try to avoid the authoring scenarios described in situations 3, 4, and 5 above. Doing so will prevent unexpected or undesirable behaviour, which can be directly attributed to modifying the definitions of International concepts. These situations are, however, permitted if they are necessary to meet legitimate clinical needs or to correct identified clinical issues.

Tip

For more information about classification please see 5.6.1.1 Classifying an Edition.

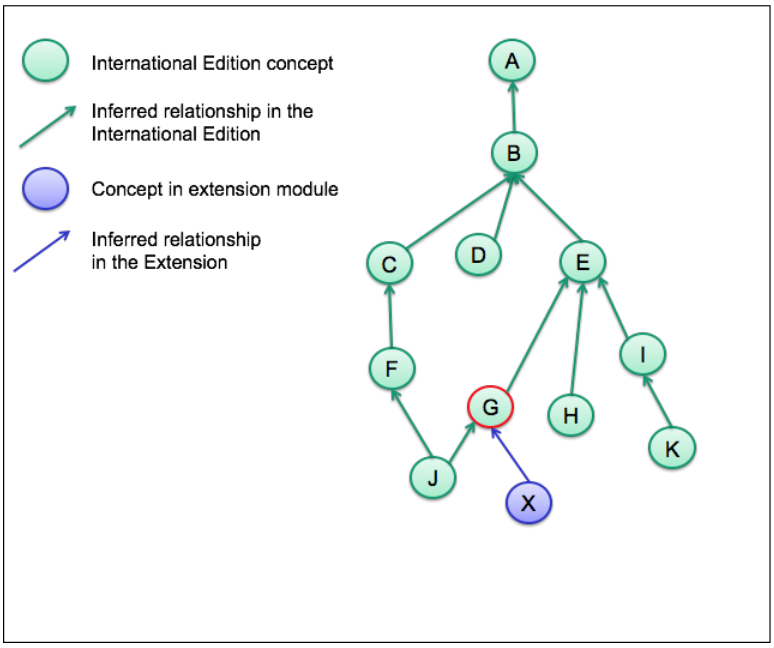
A good way to assess the impact of a change is to compare the transitive closure of subtype (|is a|) relationships after classifying the terminology with and without the extension. As a general rule, the transitive closure of the International Edition should not be modified by classifying it together with an extension.

Note

If an extension producer makes any changes or additions within their extension, which result in modification of the transitive closure of the International Edition, they must report this to SNOMED International. This will allow the issue that necessitated the change to be assessed and where appropriate resolved in the Internation Edition.

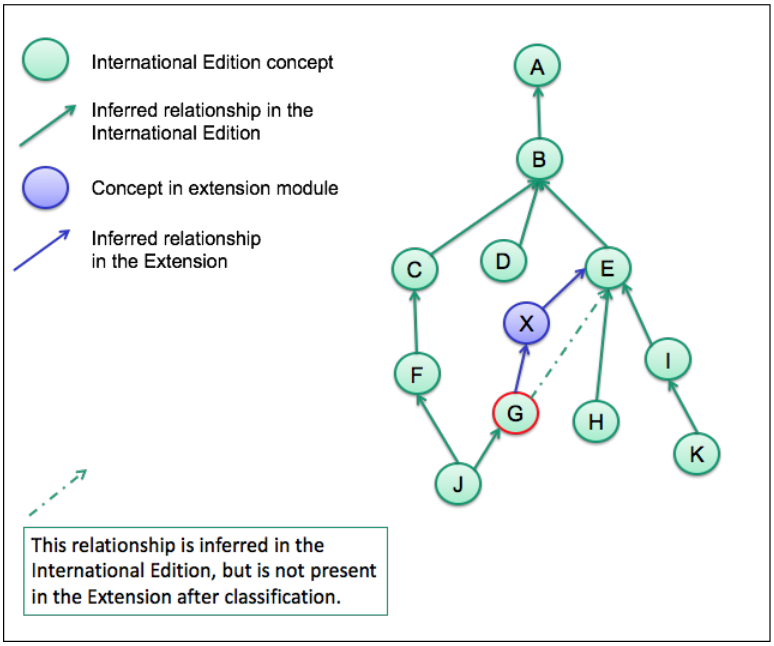
Examples
The following 3 examples illustrate the impact of adding extension concepts as leaf concepts or intermediate concepts (after classification). The impact is considered in terms of how each addition effects subsumption testing using an example international concept 'G'.

Table 5.4.2.1-1: Example 1 - Primitive Extension Concept as a Leaf Concept

<table>
<tr><td>Illustration</td><td>
SNOMED CT hierarchy after adding extension concept X


Impact of adding extension concept X on subsumption for G
Excerpt from transitive closure International Edition
(subtypes or supertypes of G)
<table>
<tr><th>Subtype</th><th>Supertype</th></tr>
<tr><td>G</td><td>E</td></tr>
<tr><td>G</td><td>B</td></tr>
<tr><td>G</td><td>A</td></tr>
<tr><td>J</td><td>G</td></tr>
</table>
Excerpt from transitive closure of International Edition and extension
(subtypes or supertypes of G)
<table>
<tr><th>Subtype</th><th>Supertype</th></tr>
<tr><td>G</td><td>E</td></tr>
<tr><td>G</td><td>B</td></tr>
<tr><td>G</td><td>A</td></tr>
<tr><td>J</td><td>G</td></tr>
<tr><td>X</td><td>G</td></tr>
</table>
</td></tr>
<tr><td>Impact on hierarchy</td><td>A primitive extension concept, which is created as a leaf concept, must always be a subtype of an international edition concept, and will not be classified as a supertype of an international concept.</td></tr>
<tr><td>Impact on subsumption</td><td>The addition of this type of content does not impact subsumption testing of concepts in the international edition, because it will remain distal in the hierarchy to all International content after classification.
In this case, the transitive closure of the International Edition will stay the same, except for the addition of the new inferred |is a| relationship associated with the concept from the extension.</td></tr>
</table>

Table 5.4.2.1-2: Example 2 - Extension Concept as an Intermediate Concept (No impact on subsumption)

<table>
<tr><td>Illustration</td><td>
SNOMED CT hierarchy after adding extension concept X


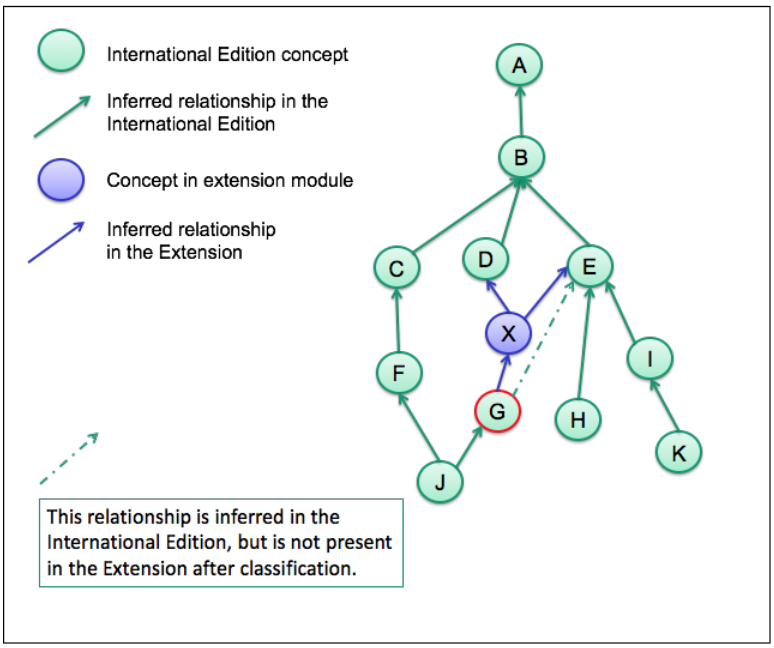
Impact of adding extension concept X on subsumption for G
Excerpt from transitive closure International Edition
(subtypes or supertypes of G)
<table>
<tr><th>Subtype</th><th>Supertype</th></tr>
<tr><td>G</td><td>E</td></tr>
<tr><td>G</td><td>B</td></tr>
<tr><td>G</td><td>A</td></tr>
<tr><td>J</td><td>G</td></tr>
</table>
Excerpt from transitive closure of International Edition and extension
(subtypes or supertypes of G)
<table>
<tr><th>Subtype</th><th>Supertype</th></tr>
<tr><td>G</td><td>E</td></tr>
<tr><td>G</td><td>B</td></tr>
<tr><td>G</td><td>A</td></tr>
<tr><td>G</td><td>X</td></tr>
<tr><td>J</td><td>G</td></tr>
</table>
</td></tr>
<tr><td>Impact on hierarchy</td><td>The definition of a concept in an extension module may result in it being classified as an intermediate concept in the SNOMED CT polyhierarchy, as illustrated in the diagram above.

This scenario results in the creation of two new inferred |is a| relationships in the extension, i.e. the relationships G | is a | X and X | is a | E. The inferred relationship G | is a | E (green dotted line), which is present in the international edition, becomes redundant when classified together with the extension (therefore marked grey in the diagram).</td></tr>
<tr><td>Impact on subsumption</td><td>This type of intermediate extension concept does not change the results of subsumption testing between international concepts. It does, however, have an impact on which relationships from the international edition are non-redundant in the extension edition. In this example, the inferred relationship G |is a| E becomes redundant when combined with the inferred relationships from the extension.

Extension producers can handle this redundancy by:

• Inactivating the inferred relationship from the International Edition, i.e. inactivate the relationship G | is a | E. For more information, see 5.4.4.3 Inactivate Relationship in an Extension.</td></tr>
</table>

Table 5.4.2.1-3: Example 3 - Extension Concept as an Intermediate Concept (Impact on subsumption)

Illustration	**SNOMED CT hierarchy after adding extension concept X**  **Impact of adding extension concept X on subsumption for G** Excerpt from transitive closure International Edition *(subtypes or supertypes of G)* **Subtype** / **Supertype**: G/E; G/B; G/A; J/G Excerpt from transitive closure of International Edition and extension *(subtypes or supertypes of G)* **Subtype** / **Supertype**: G/E; G/B; G/A; ***G*** / ***X***; ***G*** / ***D***; J/G
Impact on hierarchy	The definition of a concept in an extension module may result in it being classified as an intermediate concept in the SNOMED CT polyhierarchy , as illustrated in the diagram above . This scenario results in the creation of three new inferred |is a| relationships in the extension, i.e. the relationships **G | is a | X** and **X | is a | D and X | is a | E**. The inferred relationship **G | is a | E** (green dotted line), which is present in the international edition, becomes redundant when combined with the inferred relationships from the extension. Furthermore, the definition of the extension concept **X** results in a modification of the definition of the international concept **G**, as **G** is now a subtype of the international concept **D.**
Impact on subsumption	This type of intermediate extension concept may change the results of subsumption testing for particular international concepts, when used within the local edition. Extension producers should therefore exercise extreme caution when introducing this type of concept addition. When using the International Edition on its own, the concept **G** will **not** be treated as a subtype of the concept **D**. However, users of this example extension edition will see the international concept **G** as a subtype of the international concept **D**. This means that queries over international concepts stored in clinical data will lead to different results, depending on which edition is used. Therefore, intermediate concepts of this type may have serious consequences on the comparability and interoperability across SNOEMD CT Editions.

Process

Concepts added to an extension are represented in a concept file. These concepts are part of a module which includes the namespace identifier assigned to the extension producer by SNOMED International. Each concept also requires additional components and reference set members to be defined. At a minimum, the following components and derivatives should be created:

- The concept component, which represents the actual clinical meaning required in the extension
- At least one description component of type |Fully specified name|
- At least one description component of type |Synonym|
- At least one relationship of type |is a|, which ensures that the concept is a descendant of |SNOMED CT Concept|
- A language reference set row for each new description to specify the acceptability of the description within the relevant language or dialect

The table below provides a summary of the process to follow when adding a new concept to an extension.

File Type	Process
Concept	A new concept identifier is allocated within the extension namespace.
	The attributes of the new concept are set as follows: • *id* is set to the new concept identifier allocated within the extension namespace • *effectiveTime* is set to the date the extension will be published • *active* is set to 1 to indicate that the new concept will be active at the time of publication • *moduleId* is set to the conceptId of a module that is managed by the extension producer • *definitionStatusId* is set to state whether the concept is primitive or fully defined
Description	A description of type of | Fully specified name | and at least one description of type of | Synonym | is added. For more information on creating descriptions, please refer to 5.4.3.1 Add Description in an Extension
Stated Relationship	At least one | Is a | relationship to a parent concept is added. Note that concepts may have more than one | Is a | relationship as SNOMED CT is polyhierarchical.
	Optionally, one (or more) attribute relationship is added. The principles for adding these relationships are similar to that of adding the required subtype relationship. Please refer to 5.4.4.1 Add Relationship in an Extension for more information about adding relationships.
Language Refset	The new descriptions of the concept are referenced in at least one language refset to indicate language preferences.

5.4.2.2 Modify Concept in an Extension

Purpose

The main reasons for modifying a concept in an extension are:

- The concept is marked as fully defined, but should be primitive
- The concept is marked as primitive, but has now been fully defined
- The concept is inactive but needs to be reactivated
- The concept is active, but needs to be inactivated (Note - This scenario is described further in 5.4.3.3 Inactivate Description in an Extension.)

If an extension producer needs to modify the definition of a concept, this will necessitate adding, modifying or inactivating the relationships for which the given concept is the source. For more information, please refer to 5.4.4 Authoring Relationships.

If an extension producer needs to modify the terms used to describe the concept, this will necessitate adding, modifying or inactivating the descriptions associated with the concept. For more information, please refer to 5.4.3 Authoring Descriptions.

Principles

Overview

The clinical meaning of each concept in SNOMED CT is permanent, and can not be modified over time. This clinical meaning is captured by the concept's Fully Specified Name. Changing the clinical meaning of a concept therefore requires inactivating the concept, and creating a new concept that represents the new meaning.

The following changes, however, are permitted:

- Modifying mutable attribute values, such as definitionStatusId.
- Adding, modifying or inactivating descriptions, including changing the concept's Fully Specified Name to conform to editorial policy (as long as there is no change in clinical meaning). Please refer to 5.4.3 Authoring Descriptions.

- Changing the way that a clinical meaning is formally defined. This can be done by adding, modifying or inactivating the concept's defining relationships and/or changing the definition status of the concept. Please refer to 5.4.4 Authoring Relationships.

Modifying Mutable Attribute Values

The attributes of a concept may be modified, as long as the change is limited to the values of its mutable attributes. The values of immutable attributes should never be modified.

The effectiveTime attribute is used to support the versioning of each concept. Permitted concept modifications therefore include:

- **active**: Changing the concept's active attribute from active to inactive, or from inactive to active.
- **definitionStatusId**: Changing the concept's definition status from primitive to fully defined, or from fully defined to primitive.
- **moduleId**: Changing the concept's module. For example, this may occur when a concept is promoted. Please refer to 5.7.1.1 Moving Components Between Extensions.

It should be noted that extensions should not modify the attributes of an international concept, unless this modification is necessary to meet legitimate clinical needs or to correct identified clinical issues. Any new version of an international concept that is created by an extension producer (other than SNOMED International) and which modifies the concept's mutable attributes, should be:

- Assigned to an extension module (and *not* an international module) to reflect the fact that the modification was not made by SNOMED International.
- Submitted to SNOMED International with an explanation as to why the change was necessary.

Process

The table below provides a summary of the process to follow when modifying a concept in an extension.

File Type	Process
Concept	A new row, which represents the new version of the concept, is added to the concept file.
	The attributes of the new version of the concept are set as follows: • *id* is set to the conceptId of the concept being modified • *effectiveTime* is set to the date the extension will be published • *active* is set to indicate whether or not the concept is active at the given effective time ('1' for active and '0' for inactive) • *moduleId* is set to the conceptId of a module that is managed by the extension producer • *definitionStatusId* is set to indicate whether the concept is primitive or fully defined

5.4.2.3 Inactivate Concept in an Extension

Purpose

Concepts may be inactivated in an extension for various reasons including:

- The concept is erroneous, obsolete or out of scope
- The concept is ambiguous, and must be replaced with one or more concepts whose meaning is clear
- The concept is redundant, because another concept has the same clinical meaning or definition

Please note that concepts that are promoted from the extension into the International Edition (or a module on which the extension module depends) are **not** inactivated in the extension. For more information on concept promotion, please refer to 5.7.1.1 Moving Components Between Extensions.

Principles

Inactivating Extension Concepts

Concepts in an extension can be inactivated if necessary. This is accomplished by creating a new inactive version of the concept, and new inactive versions of any relationships in which that concept participates. This inactivation process is explained in more detail below.

Inactivating International Concepts

Concepts which belong to the International Edition (or to a module on which the extension depends) should not be inactivated in an extension. Extension producers should submit any requests for inactivation to SNOMED International (or the module owner). In most situations, in which an extension producer needs to exclude specific international concepts in their extension, this should be done by creating a reference set of either the 'included' or 'excluded' concepts. For more information, refer to 3.2.1.3. Exclude Content in the Practical Guide to Reference Sets.

Please note that if a situation arises in which an error is detected in the International Edition that causes inference errors in the extension, then this **must** be submitted to SNOMED International for correction. If a correction in the International Edition is not available prior to the release of the extension, then the error may be corrected in the Extension Edition, as long as this is reconciled in the next version of the extension that uses the corrected International Release.

Process

When inactivating a concept in an extension, key steps include:

- Inactivating the concept
- Inactivating any relationship in which the inactive concept participates
    - This includes any relationship in which the inactive concept is the source concept, the destination concept or the relationship type
- Representing the reasons for inactivation and possible replacements in the appropriate reference sets

Please note that active descriptions should **not** be inactivated when a concept is inactivated. This provides a mechanism to see the terms associated with concepts that have previously been entered into a clinical record, and to support historical queries on data that was captured using a previous version of the terminology.

> **Note**
>
> Concepts should NOT be used for data entry once they have been inactivated. However, inactive concepts should be retained to support historical records entered prior to the inactivation.

The table below provides a summary of the process to follow when inactivating a concept in an extension.

File Type	Process
Concept	A new row representing an inactivated version of the concept is added to the Concept file.
	The attributes of the new version of the concept are set as follows: • *id* is set to the conceptId of the concept being inactivated • *effectiveTime* is set to the date the extension will be published • *active* is set to '0' to indicate that the concept will become inactive at the time of publication • *moduleId* is set to identify a module in the extension • *definitionStatusId* is set to 900000000000074008 |Primitive|

File Type	Process
Stated Relationship	A new row representing an inactivated version of each active relationship, in which the concept participates, is added to the Stated Relationship file.
	An inactive concept does not participate in any active **relationships**. This means that when inactivating a concept, all active relationships, in which the concept was used as the source, destination or type, must be inactivated. As a result, the inactive concept is removed from the subtype hierarchy, and will no longer have any defining relationships. This reinforces the point that an inactive concept should not be used in any new data entry, as it will not be subsumed by any other concept. Inactivating a concept's active relationships involves adding a new row to the Stated Relationship file, with the active attribute set to '0' and the effective time set to a more recent time, for each relationship in which the inactive concept appears as the source, destination or type. For more information please refer to 5.4.4.3 Inactivate Relationship in an Extension.
Concept inactivation indicator reference set **+** **Historical association reference set**	A new row is added to the |Concept inactivation indicator attribute value reference set|, and the relevant |Historical association reference set| to indicate the reason that the concept was inactivated, and to specify any relevant associations with active concepts (e.g. possible replacements for the inactivated concept). When inactivating a concept, it is best practice is to specify the reason that the concept was inactivated in the |Concept inactivation indicator attribute value reference set|. Please refer to 4.2.3. Attribute Value Reference Set for an example of the |Concept inactivation indicator attribute value reference set|, and to 4.2.2 Component Inactivation Reference Sets for a list of valid inactivation indicator values. Additionally, depending on the inactivation reason, a row should be added to the relevant |Historical association reference set|. This helps to support extension consumers with the change management process, by (for example) specifying possible replacements for the inactivated concept. Please refer to 4.2.3 Historical Association Reference Sets for a list of historical association reference sets, and an example of the |REPLACED BY association reference set|. Please refer to 3.2.6.3. Managing Component Inactivation for further information.

5.4.3 Authoring Descriptions

Authoring descriptions in an extension may involve creating new descriptions, inactivating descriptions or modifying descriptions.

The sections that follow will examine the purpose, principles and process for each of these authoring tasks.

5.4.3.1 Add Description in an Extension

Purpose

The main reasons for adding descriptions in an extension are:

- To translate SNOMED CT into another language

- To add terms that are preferred or accepted within a local setting, for example
  - Common clinical terms that facilitate searching for concepts
  - Patient friendly terms
- To support the creation of new concepts in the extension, which each require at least two descriptions

Principles

Overview

As depicted in Figure 5.4.3.1-1, a description in an extension may refer to any concept in the same extension (Extension B), any concept in a module upon which the extension modules depend (Extension A), or any concept in the International Edition.

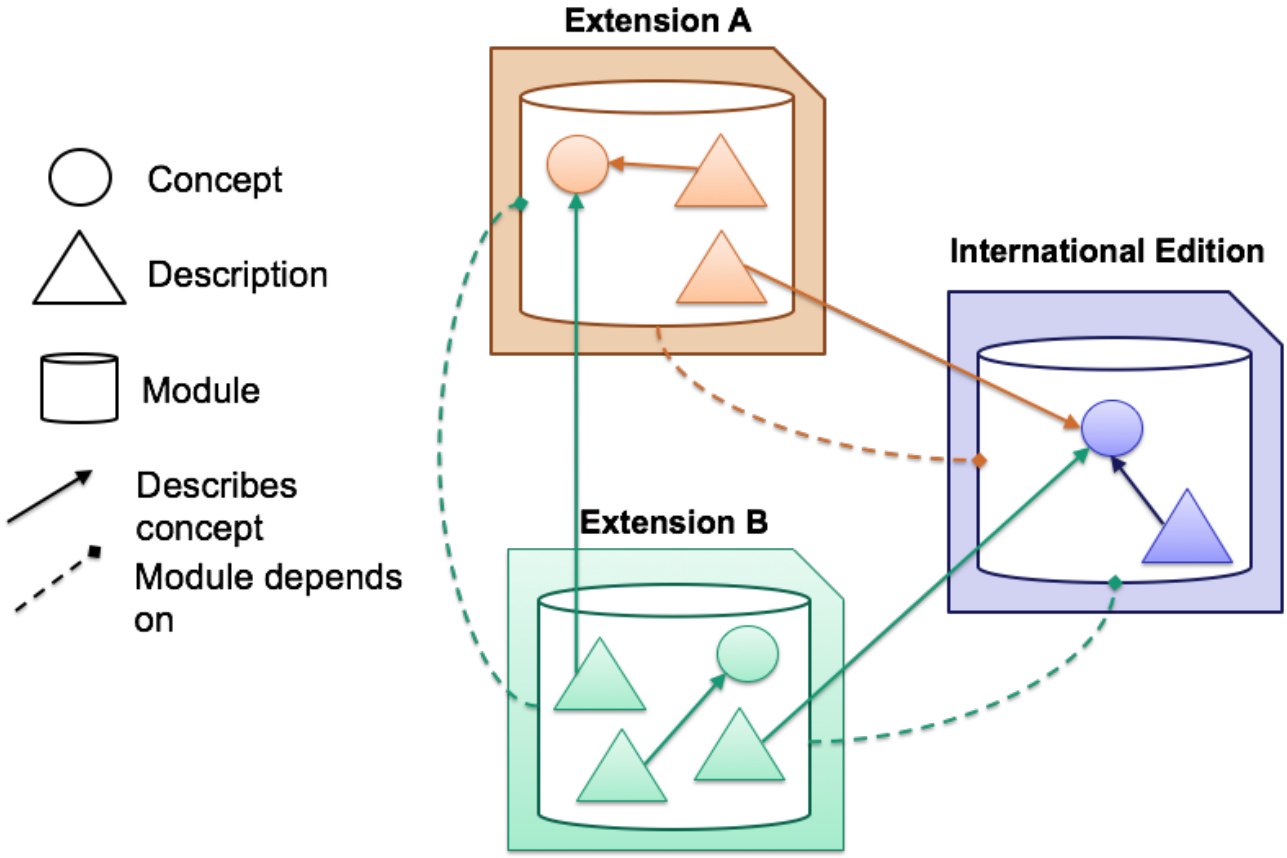


Figure 5.4.3.1-1: Descriptions can reference concepts in other modules on which the extension modules depend

Note that the green and orange triangles pointing to the purple circle represent a situation in which a description is added to an extension that describes an international concept. These extension descriptions supplement the descriptions that are already part of the International Edition.

Adding Descriptions for a New Concept

Descriptions in SNOMED CT are represented in a description file. At least two descriptions must be created for each new concept in an extension:

- A description of type |Fully specified name| (FSN)

    - Note that all extension producers should create an unambiguous FSN in US English for each new concept. Additional FSNs may also be created to support other languages and dialects.
- A description of type |Synonym|, in either English or an alternative language

The acceptability of new descriptions must be specified in a language reference set.

Adding Fully Specified Names
The following principles apply to adding FSNs in an extension.

- There may be more than one active description with a *typeId* of |Fully specified name| (FSN).
    - However, only one FSN should be marked as preferred for use in a given language or dialect by a specific Language Reference Set.
- Every extension concept must have an unambiguous FSN in US English.
    - The US English FSN is the point of reference for the meaning of all concepts in the SNOMED CT International Edition.
    - The US English FSN is used to facilitate sharing and to resolve potential issues related to the interpretation of the meaning.
- Extensions producers are permitted to create a FSN in each of their native languages.
    - This means that a non-English FSN may be marked as the preferred FSN in a specific language reference set in the extension.
    - Consequently, a concept may have more than one FSN. However, only one may be preferred in a specific language reference set.
    - Where a concept has only one active description with a *typeId* of |Fully specified name| across all language codes within a release, then that description can be assumed to be the FSN for all languages and dialects, and need not be explicitly included in every language reference set.

Adding Synonyms

- Unlike FSNs, *synonyms* are not necessarily unique between concepts, as the same *term* can be used to describe more than one concept.
- The preferred term is the synonym marked as preferred for use in the Language Reference Set for a given language or dialect.
    - There must be at least one description with a *typeId* of |Synonym| and an *acceptabilityId* value of |Preferred| for each concept associated with a description in a given language reference set.

Adding Descriptions to an Existing Concept

When a description is created in an extension, as part of a translation or to provide a localized synonym for an existing concept, a new row should be added in the relevant language reference set to indicate whether the description is |Preferred| or |Acceptable| in the given language or dialect.

Process

The table below provides a summary of the process to follow when adding new descriptions to an extension.

File Type	Process
Description	A new row which represents the new description is added to the description file.

File Type	Process
	The attributes of the new description are set as follows: • *id* is set to a new descriptionId allocated within the extension namespace • *effectiveTime* is set to the date the extension will be published • *active* is set to 1 to indicate that the new description will be active at the time of publication • *moduleId* is set to the conceptId of a module that is managed by the extension producer • *conceptId* is set to the *id* of the concept to which this description applies • *languageCode* is set to the two character code of the language in which this term was authored • *typeId* is set to indicate the type of the description • Values include |Fully specified name| or |Synonym| • *term* is set to the string of characters used to describe the given concept • *caseSignificanceId* is set to indicate the case significance of the term
Language Reference Set	A new row (or member) is added to each relevant language reference set.

	The attributes of the new language reference set member are set as follows:
	• *id* is set to a unique automatically-generated UUID
	• *effectiveTime* is set to the date the extension will be published
	• *active* is set to 1 to indicate that the new member will be active at the tiem of publication
	• *moduleId* is set to the conceptId of a module that is managed by the extension producer
	• *refsetId* is set to the conceptId of the language reference set to which the member is added
	• *referencedComponentId* is set to the descriptionId of the new description
	• *acceptability* is set to indicate the acceptability of the new description in the relevant language or dialect
	  • Value is either |Preferred| or |Acceptable|
	  • Each concept may have only one preferred FSN and one preferred synonym in each language reference set

5.4.3.2 Modify Description in an Extension

Purpose

The main reasons for modifying a description in an extension are:

- The description term contains an error that needs fixing
- The case significance of the term needs to be changed
- The acceptability of the description has changed - e.g. from acceptable to preferred, or from preferred to acceptable.

Principles

Overview

Making changes to existing descriptions requires careful consideration, because the descriptions may have been used in clinical records to represent the meaning of the associated concept. Changes are, however, permitted, as long as only mutable attributes are modified. Immutable attribute values should not be modified.

The *conceptId* field, the *languageCode* field and the *typeId* field cannot change between different versions of the same description. If a change is required to one of these immutable attributes, then the existing description should be inactivated, and a new description with the required attribute values should will be added.

Extension producers should not modify descriptions, which are part of the International Edition. When issues with international descriptions arise, SNOMED International should be notified so the issue may be addressed in a subsequent release.

Changing Mutable Description Attributes

Table 5.4.3.2-1 lists the various description attributes and their mutability. The following modifications to the mutable attributes are permitted

- Changing the *active* attribute to 0 (i.e. inactive). For more information, please refer to Inactivating a Description in an Extension.
- Changing the term of the description. Please note that only limited changes may be made to the term field, as defined by editorial rules.
    - Permitted changes include modifications that do not alter the meaning, for example a spelling correction. For a list of permitted changes, please refer to the appropriate section of the Editorial Guide:
        - FSNs - refer to Changes in the Fully Specified Name
        - Other description types - refer to capitalization
    - When in doubt, an extension producer should inactivate the description and add a new description with the required term.

Table 5.4.3.2-1: Mutability of description attributes.

Field	Data type	Purpose	Mutable	Part of Primary Key
id	SCTID	Uniquely identifies the description.	NO	YES (Full/ Snapshot)
effectiveTime	Time	Specifies the inclusive date at which the component version's state became the then current valid state of the component	YES	YES (Full) Optional (Snapshot)
active	Boolean	Specifies whether the state of the description was active or inactive from the nominal release date specified by the effectiveTime .	YES	NO
moduleId	SCTID	Identifies the description version's module. Set to a child of 900000000000443000 | Module | within the metadata hierarchy.	YES	NO
conceptId	SCTID	Identifies the concept to which this description applies. Set to the identifier of a concept in the 138875005 | SNOMED CT Concept | hierarchy within the Concept. Note that a specific version of a description is not directly bound to a specific version of the concept to which it applies. Which version of a description applies to a concept depends on its effectiveTime and the point in time at which it is accessed.	NO	NO
languageCode	String	Specifies the language of the description text using the two character ISO-639-1 code. Note that this specifies a language level only, not a dialect or country code.	NO	NO
typeId	SCTID	Identifies whether the description is fully specified name a synonym or other description type. This field is set to a child of 900000000000446008 | Description type | in the Metadata hierarchy.	NO	NO
term	String	The description version's text value, represented in UTF-8 encoding.	YES	NO

caseSignificanceId	SCTID	Identifies the concept enumeration value that represents the case significance of this description version. For example, the term may be completely case sensitive, case insensitive or initial letter case insensitive. This field will be set to a child of 900000000000447004 | Case significance | within the metadata hierarchy.	YES	NO

Changing the Acceptability of Descriptions

An extension producer may also need to change the acceptability of a description. For example, a synonym may change from being acceptable to preferred, or from being preferred to acceptable. This type of change does not require any modifications to the Description file. Instead, the associated member of the relevant language reference set must be modified. For more information, please refer to 5.4.6.2 Modify Members of a Reference Set.

Process

The table below provides a summary of the process to follow when modifying descriptions in an extension.

File Type	Process
Description	A new row which represents a new version of the description is created.
	The attributes of the new version of the description are set as follows: • *id* is set to the descriptionId of the description being modified • *effectiveTime* is set to the date the extension will be published • *active* is set to 1 to indicate that the new version of thedescription will be active at the time of publication • *moduleId* is set to the conceptId of a module that is managed by the extension producer • *conceptId* is set the same as the original version of the description • *languageCode* is set the same as the original version of the description • *typeId* is set the same as the original version of the description • *term* is set to the (possibly updated) string of characters used to describe the given concept. Note: Only limited changes, in accordance with editorial rules, can be made • *caseSignificanceId* is set to indicate the (possibly updated) case significance of the term
Language Reference Set	If no changes to the acceptability of the description are required, then no changes to the associated language reference set members are necessary. If changes to the acceptability of the description are required, a new version of the relevant language reference set member is added with the required *acceptabilityId*.

5.4.3.3 Inactivate Description in an Extension

Purpose

There are various reasons for inactivating descriptions in an extension, including:

- The description does not represent the same meaning as the fully specified name (FSN) of the concept
- The description is no longer current, useful, appropriate or acceptable
- The description fails to comply with the current editorial guidance. For more information, please refer to Terming and Naming Conventions in the Editorial Guide.
- The description contains a technical error, such as a typographic error

Principles

Excluding Extension Descriptions

Descriptions created in an extension can be inactivated if necessary. This is done by inactivating both the description itself and the associated member(s) of the relevant language reference set(s). It is recommended that the reason for inactivation is specified in the Description Inactivation Reference Set.

Excluding International Descriptions

Descriptions which belong to the International Edition (or another module on which the extension depends) should not be inactivated in the extension. If there is a requirement to remove international descriptions from an extension, this should be accomplished by excluding the given description identifiers from the relevant language reference set(s). Two possible approaches for this are described below.

Copy and Adapt Language Reference Set
This approach involves creating a copy of one of the language reference sets distributed with the International Edition, and adding or removing members as required. All members of the new language reference set must be assigned to a module in the extension. When using this approach, the extension producer is responsible for maintaining the new language reference set. If the extension producer wants their new language reference set to incorporate future changes made to the international language reference set, they are responsible for maintaining these changes.

Adopt and Adapt Language Reference Set
An extension producer may choose not to create a new copy of an international language reference set. Instead, they may use the international language reference set directly, and exclude any members of this reference set by inactivating the relevant members in their extension module. To exclude a description from an international language reference set, a new version of the reference set member is created with the following attributes:

- *id* is set to the same value as that of the member of the international language reference set
- *effectiveTime* is set to the release date of the extension
    - Note that this date should be after the effectiveTime of the reference set member in the international release
- *active* is set to '0' to indicate that the member is being inactived
- *moduleId* is set to a module in the extension
- *refsetId* is set to the same value as that of the international language reference set
- *acceptabilityId* is set to the same value as in the original member of the international language reference set

Please note that if a new version of the reference set member is published in the International Edition, this will become the current version of that member. It may, therefore be necessary to create a new inactive version in the extension with a more recent effectiveTime.

Process

This table describes the process of excluding descriptions in an extension.

File Type	Process
Description	To exclude descriptions in an extension, a new inactive version of the description is added to the description file. The attributes of the new version of the description are set as follows: • *id* is set to the descriptionId of the description being inactivated • *effectiveTime* is set to the date the extension will be published • *active* is set to '0' to indicate that the description is being inactivated • *moduleId* is set to identify a module in the extension The remaining attributes are set to the same value as in the previous version of the description.

File Type	Process
Language Reference Set	A new row, which references the description to be excluded, with the following attribute values: • *id* is set to the same value as the reference set member being excluded • *effectiveTime* is set to the date the extension will be published • *active* is set to '0' to indicate that the reference set member is being inactivated • *moduleId* is set to identify a module in the extension The remaining attributes are set to the same value as in the previous version of the reference set member.
Description inactivation indicator reference set	The |Description inactivation indicator attribute value reference set (foundation metadata concept)| represents the reason that each inactive description was inactivated. When inactivating a description, it is good practice to specify the reason for inactivation in this reference set. The |Description inactivation indicator attribute value reference set (foundation metadata concept)| allows extension consumers to understand the reason that each description was inactivated. Please visit 3.2.6.3. Managing Component Inactivation for further information.

5.4.4 Authoring Relationships

Authoring relationships in an extension may involve creating new relationships, inactivating relationships or modifying existing relationships.

The sections that follow will examine the purpose, principles and process for each of these authoring tasks.

5.4.4.1 Add Relationship in an Extension

Purpose

Relationships may be added in an extension for various reasons including:

- Specifying the defining characteristics of new extension concepts
- Improving the definition of an existing extension concept
- In exceptional cases, it may be permissible to add defining relationships to
    - Concepts which belong to the International Edition
    - Concepts which belong to another module on which the extension depends

Principles

Overview

A SNOMED CT relationship involves three main concepts - the source concept, the destination concept, and the relationship type concept (also known as the 'attribute'). Each of the concepts in an extension relationship may belong to either a module from the extension itself, or a module upon which the extension depends (e.g. an international module).

When adding new relationships in an extension, the following principles apply:

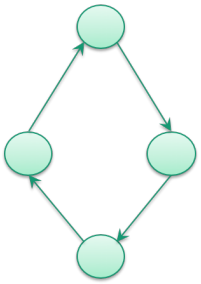
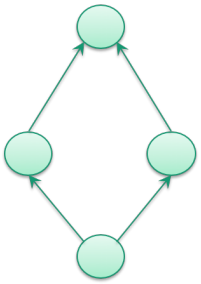
- **Source Concept**: Except in exceptional cases (noted below), the relationship must represent a defining characteristic of a concept in a module for which the extension producer is responsible.
    - This means that the sourceId of the relationship should refer to a concept in the extension.

    > In cases in which it is necessary to meet legitimate clinical needs or to correct identified clinical issues, an extension producer may add additional defining relationships to a concept which belongs to an International module (or a module on which the extension module depends). However, this should be done with extreme caution due to the potential impact on the classification results. If these situations arise, SNOMED International (or the module owner) must be notified.

- **Attribute**: The type of relationship should usually be represented by an attribute concept which is part of the SNOMED International Edition.
    - When using an attribute from the International Edition, concept model rules and editorial guidance should be followed. For example, relationships should comply with the rules stated in the MRCM.
    - In some cases, new attribute concepts may be added to an extension. While this is permitted, new attributes should be applied with caution, and concept model rules and editorial guidance should be clearly documented by the extension provider.

    > Attributes added in an extension can lead to inconsistent classification, where attributes overlap with or interact with attributes in the SNOMED International Edition. Furthermore, attributes created in an extension may not be well aligned with new attributes covering similar characteristics that may be included in future releases of the International Edition. Therefore, care should be exercised when adding attributes in an extension. In particular, the rationale for creating each new attribute should be well documented, with clear guidance on its consistent use. Ideally additions should also be made to the MRCM reference sets to capture the associated concept model rules in a machine processable way. If similar attributes are later added to the International Edition, the extension producer should plan to align with the International Edition as soon as possible.

- **Destination Concept**: The target of the relationship may belong to the extension module, or any module on which the extension module depends (including an international module). However, care should be taken to avoid intermediate concepts as described in 5.4.2.1 Add Concept in an Extension). Additionally, it is important to ensure that the added relationships do not induce any cycles, i.e. it should be ensured that the added relationships retain the SNOMED CT hierarchy as a Directed Acyclic Graph.

Cyclic Graph	Directed Acyclic Graph
NOT ALLOWED	ALLOWED
	

Process

The table below provides a summary of the process to follow when adding a new relationship to an extension.

File Type	Process
Stated Relationship	A new row representing a new relationship is added to the stated relationship file.
	A new relationship identifier is allocated within the extension namespace.
	The attributes of the new relationship are set as follows: • *id* is set to the new relationship identifier allocated within the extension namespace. • *effectiveTime* is set to the date the extension will be published • *active* is set to '1' to indicate that the new relationship will be active at the time of publication • *moduleId* is set to identify a module concept from the extension • *sourceId* is usually set to a concept from an extension module. • Concepts from other modules upon which the extension is dependent may be used if necessary to meet legitimate clinical needs or to correct identified clinical issues. • *destinationId* is set to a concept that represents the value of the relationship attribute • *relationshipGroup* is set to a number that indicates which relationships with the same sourceId are logically grouped together • *typeId* is set to an attribute concept that represents the type of the relationship • *characteristicTypeId* is usually set to 900000000000010007 |Stated relationship| • *modifierId* is usually set to 900000000000451002 |Existential restriction modifier|
Relationship	Once authoring is complete, the contents of the extension modules, together with every module on which these depend, are classified. The resulting set of inferred relationships (i.e. the output of the classification process) is added to the Relationship file.
	A new row is added to the Relationship file for every inferred relationship that results from the classification process.
	The attributes of the inferred relationships are set as follows: • *id* is set to a new relationship identifier allocated within the extension namespace. • *effectiveTime* is set to the date the extension will be published • *active* is set to '1' to indicate that the new relationship will be active at the time of publication • *moduleId* is set to identify a module concept from the extension • *sourceId* is set to the source concept in the relationship. This will usually belong to the extension module. • *destinationId* is set to the destination concept in the relationship • *relationshipGroup* is set to a number that indicates which relationships with the same sourceId are logically grouped together • *typeId* is set to an attribute concept that represents the type of the relationship • *characteristicTypeId* is usually set to 900000000000011006 |Inferred relationship| • *modifierId* is usually set to 900000000000451002 |Existential restriction modifier|

5.4.4.2 Modify Relationship in an Extension

Purpose

There are a number of reasons for modifying a relationship in an extension, including:

- Changing the status of the relationship from active to inactive, or from inactive to active
- Changing the relationship group of a specific relationship
- Changing the characteristic type of a relationship

Principles

Relationships that are used to define extension concepts can be modified. However, please note that only the values of mutable attributes may be modified. As indicated in Table 5.4.4.2-1, the six mutable attributes of a relationship are: *effectiveTime, active, moduleId, relationshipGroup, characteristicTypeId,* and *modifierId*. When a change is required to the source, destination or type of the relationship, the relationship must be inactivated and a new relationship created with the required values.

All modifications to a SNOMED CT relationship should also conform to the SNOMED CT concept model rules, including relationship group cardinality constraints. For more information please refer to the Editorial Guide and the Machine Readable Concept Model.

Relationships authored by SNOMED International, which define concepts in the International Edition, should not be modified. If an error is detected in the International Edition, which requires one or more relationships to be modified, the issue should be documented and reported to SNOMED International using the Content Request Service. In situations in which an extension producer needs to modify a relationship to temporarily fix an issue, this should be done in a module owned by the extension producer, and reported to SNOMED International It is recommended that extension producers try to avoid modifying international relationships, as such modifications may affect the results of subsumption testing and therefore may have an impact on cross-edition interoperability.

Table 5.4.4.2-1: Mutability of relationship attributes

Field	Data type	Purpose	Mutable	Part of Primary Key
id	SCTID	Uniquely identifies the relationship.	NO	YES (Full/ Snapshot)
effectiveTime	Time	Specifies the inclusive date at which the component version's state became the then current valid state of the component.	YES	YES (Full) Optional (Snapshot)
active	Boolean	Specifies whether the state of the relationship was active or inactive from the nominal release date specified by the effectiveTime field.	YES	NO
moduleId	SCTID	Identifies the relationship version's module. Set to a child of 900000000000443000 | Module | within the metadata hierarchy.	YES	NO
sourceId	SCTID	Identifies the source concept of the relationship version. That is the concept defined by this relationship. Set to the identifier of a concept.	NO	NO

destinationId	SCTID	Identifies the concept that is the destination of the relationship version. That is the concept representing the value of the attribute represented by the typeId column. Set to the identifier of a concept. Note that the values that can be applied to particular attributes are formally defined by the SNOMED CT Machine Readable Concept Model.	NO	NO
relationshipGroup	Integer	Groups together relationship versions that are part of a logically associated relationshipGroup. All active Relationship records with the same relationshipGroup number and sourceId are grouped in this way.	YES	NO
typeId	SCTID	Identifies the concept that represent the defining attribute (or relationship type) represented by this relationship version. That is the concept representing the value of the attribute represented by the typeId column. Set to the identifier of a concept . The concept identified must be either 116680003 | Is a | or a subtype of 410662002 | Concept model attribute | . The concepts that can be used as in the typeId column are formally defined as follows: 116680003 | is a | OR < 410662002 | concept model attribute | Note that the attributes that can be applied to particular concepts are formally defined by the SNOMED CT Machine Readable Concept Model.	NO	NO
characteristicTypeId	SCTID	A concept enumeration value that identifies the characteristic type of the relationship version (i.e. whether the relationship version is defining, qualifying, etc.) This field is set to a descendant of 900000000000449001 | Characteristic type | in the metadata hierarchy.	YES	NO
modifierId	SCTID	A concept enumeration value that identifies the type of Description Logic (DL) restriction (some, all, etc.). Set to a child of 900000000000450001 | Modifier | in the metadata hierarchy. Currently the only value used in this column is 900000000000451002 | Some | and thus in practical terms this column can be ignored. For further clarification please see Notes on modifierId.	YES	NO

Process

The table below provides a summary of the process to follow when modifying a relationship in an extension.

File Type	Process
Stated Relationship or Relationship	A new row which represents a new version of the modified relationship is added to the respective relationship file.

File Type	Process
	The attributes of the new version of the relationship are set as follows: • *id* is set to the relationshipId of the relationship being modified • *effectiveTime* is set to the date the extension will be published • *active* is set to '1' to indicate that the new version of the relationship will be active at the time of publication • *moduleId* is set to identify a module concept from the extension • *sourceId* is set to the same as the original version of the relationship • *destinationId* is set to the same as the original version of the relationship • *relationshipGroup* is set to the updated relationship group number (or the original number if no update is required) • *typeId* is set to the same as the original version of the relationship • *characteristicTypeId* is set to the updated value (or the original value if no update is required) • *modifierId* is set to the updated value (or the original value if no update is required)

5.4.4.3 Inactivate Relationship in an Extension

Purpose

There are a variety of reasons for inactivating a relationship in an extension, including:

- An extension concept has been inactivated, which requires all relationships that it participates in to be inactivated
- To correct errors in the definition of a concept
- To remove any redundant |is a| relationships, generated when classifying the extension edition
- To remove any 'temporary' relationships that were added to correct errors in the International Edition that have since been fixed in the International release

Principles

Inactivating Extension Relationships

Relationships in an extension can be inactivated if required. This is done by creating a new version of the extension relationship with a more recent *effectiveTime* and the *active* attribute set to '0' (for 'inactive').

Inactivating Relationships from the International Release

Relationships which belong to the International Edition (or to a module on which the extension depends) should *not* be inactivated in an extension. When relationships from the International Edition need to be excluded, this can be done by creating a simple reference set of either 'included' or 'excluded' relationships.

The only situations in which an extension producer may inactivate a relationship specified in the international edition (or a module on which the extension depends) are:

- Where an international |is a| relationship becomes redundant after classification is performed. For more information, refer to 5.4.2.1 Add Concept in an Extension and 5.6.1.1 Classifying an Edition.
- Where there exists an erroneous relationship in the International Edition (or in a module on which the extension depends), that may cause incorrect inferences in the extension edition. If this occurs, the error may need to be temporarily corrected in the extension edition. In addition, SNOMED International must be notified of the error so that it can be permanently corrected in the international edition.

In these situations, the relationship from the International Edition (or module on which the extension depends) is inactivated by creating a new version of the relationship in the extension module with a more recent *effectiveTime* and the *active* attribute set of '0' (for 'inactive').

Process

The table below provides a summary of the process to follow when inactivating a relationship in an extension.

File Type	Process
Stated Relationship or Relationship	A new row representing a new version of the relationship being inactivated is added to the file.
	The attributes of the new relationship version are set as follows: • *id* is set to the same relationship identifier as the relationship being inactivated • *effectiveTime* is set to the date the extension will be published • *active* is set to 0 to indicate that the relationship will become inactive at the time of publication • *moduleId* is set to identify a module in the extension • sourceId, *destinationId*, *typeId*, *characteristicTypeId*, *typeId* are set as per the previous version of this relationship

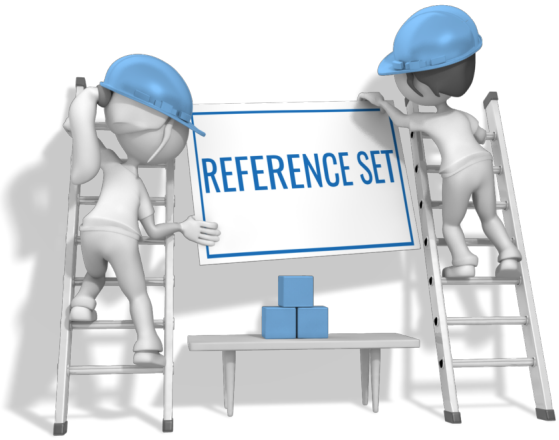
5.4.5 Authoring Reference Sets

Authoring reference sets in an extension may involve adding, modifying or inactivating members of a reference set, or the reference set itself.

The sections that follow will examine the purpose, principles and process for each of these authoring tasks.

5.4.5 Authoring Reference Sets

5.4.6 Authoring Reference Set Members



5.4.5.1 Create New Reference Set in an Extension

Purpose

Reference sets can be added in an extension to meet specific national or local requirements. Reasons for adding a reference set include:

- Specifying a subset of components for inclusion or exclusion
- Mapping between SNOMED CT and other code systems to support data integration or communication
- Specifying the acceptability of descriptions in a given language or dialect

For more information about a range of purposes for reference sets, please refer to the Practical Guide to Reference Sets.

Principles

When creating a new reference set it is important that the reference set metadata concept (which identifies and names the reference set) and the associated reference set members belongs to a module of the reference set producer. I.e. if a reference set is created as part of an extension, the reference set metadata concept should belong to a module within the namespace of the extension producer.

All reference sets follow a logical model. Each reference set type is represented by a specific data structure which enables a specific functionality. SNOMED International specifies a set of reference set types, which can be used to support a set of terminology management purposes. SNOMED CT Members (National Release Centers) and Affiliates may also create reference sets to assist with localization and effective implementation of SNOMED CT. In general, an extension producer can create a reference set within their extension, but the reference set type should be defined in either

- The producers extension
- The International Edition
- Other extensions (which the producers extension modules are dependent on)

For more information about the different reference set types and their data structures, please refer to the reference set release file specification.

Please note that for all reference sets, there should be guidance on what is possible and what is not possible. For some International reference sets, it is acceptable and even required to add extension rows to the International reference set. For example, the Module Dependency Refset, the MRCM Module Scope Refset, and Language Refset. This is not necessarily applicable to all reference sets. Over time, further guidance in this area will be developed.

Process

The table below provides a summary of the process to follow when creating a new reference set.

File Type	Process
Define the reference set in the metadata hierarchy Following steps should be taken for creating the reference set concept:	
Concept file	1. Create a concept for the reference set
Description file	2. Add up to three Descriptions for the reference set concept. I.e. the FSN, the Preferred Term and optionally the purpose, see 4.1.3. Naming Conventions for Reference Sets
Language reference set	3. Specify the acceptability of the descriptions in the applied Language Reference Set
Relationship file	4. Add an | is a | Relationship to link the reference set to the appropriate pattern
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. You may add new concepts for some of the attributes, and reuse existing concepts for other attributes, if you wish. Following steps should be taken for each attribute that you wish to create:	
Concept	1. Add a concept for the attribute
Description	2. Add Descriptions for each of the new attribute.
Language Reference Set	3. Specify the acceptability of the descriptions in the applied Language Reference Set
Relationship	4. Link the attribute with an | is a | Relationship into the 900000000000457003 |Reference set attribute (foundation metadata concept)| hierarchy

File Type	Process
⚠ If new attribute values need to be created these should also be added as SNOMED CT concepts and placed as subtypes of the concept 900000000000457003 |Reference set attribute (foundation metadata concept)|, following the process described above.	
Create the Descriptor for the reference set If the reference set does not follow an existing reference set pattern, the additional attributes specific for this customized reference set should be included in the Descriptor reference set.	
Reference Set Descriptor Reference Set	A new reference set member is created in the reference set Descriptor for each attribute in the reference set pattern. For guidance on creating a reference set member, please see 6.3.3.1 Add Members to a Reference Set. For further introduction to the reference set Descriptor, please see 4.1.1. Reference Set Descriptor in Practical Guide to Reference Sets.
Add members to the reference set Reference set members are added to the reference set, which includes specifying the attribute values for each reference set member, see 6.3.3.1 Add Members to a Reference Set.	

For additional information, see the practical guide to reference sets: 13.1 How to create a new Reference Set using an existing pattern

5.4.5.2 Modify Reference Set in an Extension

Purpose

Depending on the type of reference set, there may be different reasons for modifying the reference set, including:

- Promoting the reference set to a parent module, for example the International Edition
- Transfer of responsibility for maintenance to an organization that is responsible for a module on which the current module depends, see 6.3.2.3 End Support for a Reference Set

Principles

Principles for modifying reference sets include:

- Reference sets can be modified by changing the value of the *active* attribute or the value of the *moduleId* attribute, i.e. for activating or inactivating the reference set, or moving the reference set to another module (a module on which the current module depends).
    - The reference set metadata concept is always primitive and the concept can therefore not be changed from primitive to fully defined, or from fully defined to primitive, i.e. the value of the *definitionStatusId* attribute should be primitive for all versions of the reference set
- The reference set type should **not** be changed, i.e. the |is a| relationship, linking the reference set to the appropriate reference set type concept should not be changed
    - The only situation where the the reference set type may change is if the reference set was originally misplaced in the SNOMED CT hierarchy.
- If the reference set needs to be changed to a different type, (for example, from a simple reference set to an ordered component type reference set) then it would require the original reference set to be inactivated and a new reference set to be created in accordance with the Descriptor for that reference set type.
- Reference sets that belong to a module which is not managed by the extension producer should not be modified.

Process

Modifying a reference set created within the extension module

The process for modifying the reference set concept, is similar to the general process for modifying concepts in SNOMED CT. Therefore, please refer to 5.4.2.2 Modify Concept in an Extension for further guidance.

5.4.5.3 End Support for a Reference Set

Purpose

There are three distinct situations in which an extension producer may end support for a reference set in an extension.

- Responsibility for maintenance of a reference set is transferred to another extension producer or to SNOMED International
    - For example, a reference set currently maintained as part of a National extension may be transferred to SNOMED International if it has recognized international value.
- A decision is made to stop maintaining a reference set, without transferring the responsibility to another organization
    - For example, an organization may no longer have the resources to maintain a reference set that it created, although the reference set is still considered useful.
- A decision is made to deprecate use of an existing reference set
    - For example, if a reference set is no longer relevant, has been superseded by a more useful reference set, or is considered inaccurate or misleading in some way.

Principles

When the owner of a reference set wants to end support for the reference set, this should be indicated by making the changes described in the process section below.

In the situation where the reference set belongs to an extension, the owner of the extension may only make these changes to a reference set that is currently in an extension module for which it is responsible. The exception to that rule is that, in the case where a responsibility for maintenance of a reference set is transferred to another organization, the organization to which responsibility is transferred is required to take some of these steps.

Prior to ending support for a reference set, it is important that the reference set producer has an overview of the extent to which the reference set is used. If a producer and owner of a reference set continues to distribute an unsupported reference set with active members, there is an inherent risk that it will continue to be used. However, deprecation formally inactivates the references set members to mim this possibility.

Process

The table below provides details and considerations on the process of inactivating a reference set.

Reason for Ending Support	Organization responsible currently responsible for the reference set	Organization accepting transfer of responsibility for the reference set
Transfer of responsibility for maintenance to an organization that is responsible for a module on which the current module depends	1. Request the newly responsible organization to issue new versions of the reference set concept, metadata and active reference set members using the same identifiers but changing the moduleId to refer to the module in which the reference set is now being maintained. 2. Inform users that the reference set is now being maintained by another organization. 3. Do not inactivate or otherwise alter any of the existing concepts or reference set members. 4. Ensure that documentation explaining this change is prepared and distributed to all users of the extension.	1. Create new versions of the reference set concept and related metadata. The new versions of components must have the updated effectiveTime for the relevant release date and the moduleId of the module in which the reference set will now be maintained. However, the id and all other fields must have the same values as in the original versions. 2. Create new versions of all active members from of the reference set. The new versions of references set members must an updated effectiveTime for the relevant release date and the moduleId of the module in which the reference set will now be maintained. However, the id, refsetId and all other fields must have the same values as in the original versions.

Transfer of responsibility for maintenance to an organization that is not responsible a module on which the current module depends	1. Request the newly responsible organization to create and maintain a new reference set in their own extension module. The newly created reference set concept and metadata should replicate the and includes all all the members of the original reference set that were active immediately prior to deprecation. 2. Inactivate the original reference set concept and metadata following the process described in 5.4.2.3 Inactivate Concept in an Extension. 3. Inactivate all the members of the original reference set following the process described in 6.3.3.3 Remove Members from a Reference Set. 4. Add a row to the | Concept inactivation indicator reference set | indicating the reason for inactivation of the reference set. 5. Ensure that documentation explaining this change is prepared and distributed to all users of the extension.	1. Create a new reference set following the process described in 6.3.2.1 Create a New Reference Set. The reference set must have a newly allocated id, the updated effectiveTime for the relevant release date and the moduleId of the module in which the reference set is being created. However, the name and all other fields should have the same values as in the original versions. 2. Create new members of the newly created reference set following the process described in 6.3.3.1 Add Members to a Reference Set. A member must created matching each of the active members of original reference set. 3. Create a row in the | REPLACED BY association reference set | indicating that the original reference set has been replaced by this reference set.
Ending maintenance of without formally deprecating continued use of a reference set	1. Inactivate the reference set concept, following the process described in 5.4.2.3 Inactivate Concept in an Extension. 2. Do not inactivate the members of the reference set. 3. Add a row to the | Concept inactivation indicator reference set | indicating the reason for inactivation of the reference set. 4. Ensure that documentation explaining this change is prepared and distributed to all users of the extension.	-
Deprecating continued use of a reference set	1. Inactivate the reference set concept and metadata following the process described in 5.4.2.3 Inactivate Concept in an Extension. 2. Inactivate all the members of the reference set following the process described in 6.3.3.3 Remove Members from a Reference Set. 3. Add a row to the | Concept inactivation indicator reference set | indicating the reason for inactivation of the reference set. 4. Ensure that documentation explaining this change is prepared and distributed to all users of the extension.	-

Distribution of an Inactivated Reference Set and its Members

If the extension producer wants to avoid users from needing to import a deprecated or transferred reference set in future releases, the inactivated reference set may be separated from the main extension release (e.g. it could be in a separate release package, or accessible via a separate service or from a static location). Changes in packaging must be formally notified to users of the extension in advance of the change.

Warning

It is essential that the inactive reference set concept, metadata and reference set members are included in the first release of the original release package after the changes are made. Otherwise users applying delta updates will not be aware that the change has been made.

5.4.6 Authoring Reference Set Members

Reference set members may also be added, modified or inactivated in an extension, to meet evolving requirements.

The sections that follow will examine the purpose, principles and process for each of these authoring tasks.

5.4.6.1 Add Members to a Reference Set

Purpose

Reference set members can be added within an extension for a variety of reasons, including:

- To add members to a subset in the extension that is used nationally or locally
- To adapt a reference set created by another organization, by adding new members that are required nationally or locally

Principles

Specifying members of a reference set can be done in different ways. It will depend on the requirements for the reference set what approach is feasible and possible. Please refer to the Practical Guide to Reference Sets for detailed instructions on the approaches and methods for identifying the SNOMED CT components that will be referenced by the members of the reference set.

However, each reference set member will be represented in the reference set in accordance with the following principles and process.

Reference set members may be added to reference sets belonging to modules in

- The producers extension
- The International Edition
- Other extensions (which the producers extension modules are dependent on)

Reference set members created in the extension should be created within the module of the extension producer, so that it is possible to distinguish the reference set members created within the producers extension from reference set members created by other organisations (i.e. belonging to other modules).

Referenced Components

Even though the individual reference set members belong to the module of the extension producer, the actual components that are referenced by the reference set member, may belong to modules in

- The International Edition
- The extension in which the reference set and its members are produced
- Other extensions (which the producers extension modules are dependent on)

Process

The table below provides a summary of the process to follow when adding members to a reference set.

File Type	Process
Reference Set	A new refset row is created with a unique *id*. The data type for this id is UUID and the id can be generated using a UUID generator. Note that a namespace identifier will not be part of this id.
	Versioning and module identification attributes are set accordingly: • *effectiveTime* is set to the date the extension will be published • *active* is set to reflect the status of the reference set member, i.e. 1 for active • *moduleId* is set to identify a module concept managed by the extension producer
	Attributes common for all reference set types are set accordingly: • *refsetId* is set to the identifier of the concept used to identify and name the reference set. Note, the value of *refsetId* will be the same for all members in the reference set • *referencedComponentId* is set to the identifier of the SNOMED CT component or reference set, which is referenced by this specific reference set member
	Attributes specific to the reference set type are set. For more information, please refer to the release file specification.

5.4.6.2 Modify Members of a Reference Set

Purpose

There are a number of reasons for modifying members of a reference set in an extension (depending on the type of reference set), including:

- Updating map records to
  - Refine map rules or map advice
  - Change the target of a map
- Changing the acceptability of a description in a language reference set
- Changing the order of subset members specified in an Ordered type reference set

Principles

Principles for modifying reference sets include:

- Reference sets can be modified by
  - Adding or inactivating reference set members. Please refer to the guidance on 6.3.3.1 Add Members to a Reference Set or 6.3.3.3 Remove Members from a Reference Set.
  - Modifying mutable attribute values of reference set members. To see what attributes are mutable for each reference set type, please refer to the specification of the specific reference set type in the reference set release files specification. If the reference set type is a locally defined reference set, please consider any mutability constraints on the individual attributes.
- Don't modify immutable attributes of a reference set.
  - In the case where the modification of an immutable attribute is required, this should be done by inactivating the reference set member and creating a new reference set member with the required, updated values.
- If circumstances require you to modify reference set members that belong to another module than the producers extension, following options exist:

- Inactivating the specific reference set member in your own module and create a new reference set member with the updated value
    - The benefit of this approach is that you retain the definition and representation of the reference set member as it was intended by its original authors, and the new reference set member will be easily identified as a local reference set member, as the identifier of that reference set member is not available in the original reference set
- Create a new version of the specific reference set member in your own module, and make the necessary modifications

Proces

The table below provides a summary of the process to follow when modifying an existing member in a Reference Set.

Table 6.3.3.2-1: Modify reference set member

File Type	Process
Concept	The metadata concept representing the reference set is retained
Reference Set	A new reference set row is created and the *id* is retained from the previous version of the refset member.
	Versioning and module identification attributes are set accordingly: • *effectiveTime* is set to the date the extension will be published • *active* is set to reflect the status of the reference set member, i.e. '1' for active and '0 'for inactive • *moduleId* is set to identify a module managed by the extension producer
	Attributes common for all reference set types are set accordingly: • *refsetId* is retained as the value from the previous version of this refset member. A member cannot move from one reference set to another • referencedComponentId id retained as the value from the previous version of this refset member. A member cannot change the component which it refers to. • In this case, the existing member record should be inactivated, and a new one created.
	Attributes specific to the reference set type are set accordingly: • *additional attributes* - may be updated with a value, of type (and possibly range) limited by the descriptor record for this Reference Set attribute

5.4.6.3 Remove Members from a Reference Set

Purpose

Reference set members may be inactivated when

- The reference set member is no longer relevant or valid in the reference set in which it is being used
- The reference set to which the reference set member belongs is inactivated

Principles

Any inactivation of reference set members should be done in the module of the reference set producer.

Process

The table below provides a summary of the process to follow when removing members from a reference set.

File Type	Process
Reference Set	**The following steps should be taken to inactivate a reference set member** Create a new version of the reference set member to be inactivated. Versioning and module identification attributes are set accordingly: • *effectiveTime* is set to the date the extension will be published • *active* is set to reflect the status of the reference set member, i.e. 0 for inactive • *moduleId* is set to identify a module managed by the extension producer All other attribute values are retained.

5.5 Review and Validation

Validating the content in an extension is the process of ensuring that all components and reference sets within the extension comply with the authoring principles. As illustrated below, this process involves three main steps: Validation during authoring, post-authoring review, and pre-release validation.

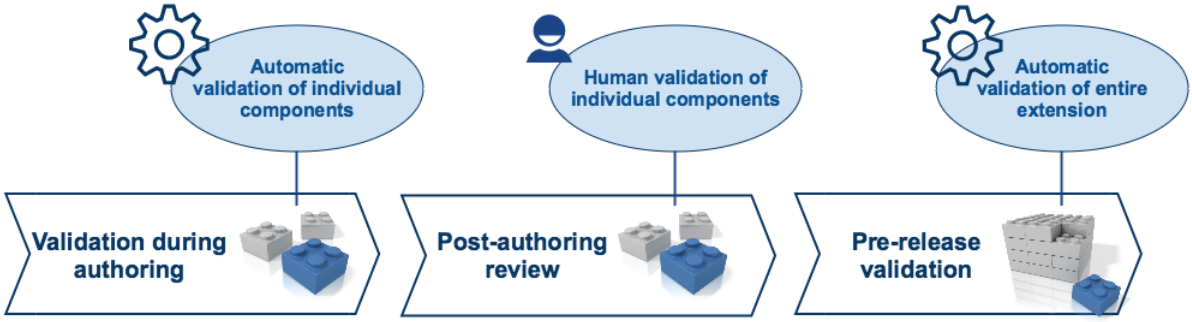


Figure 5.5-1: Extension validation involves both automated and manual processes

Effective and high quality terminology authoring processes should include thorough automated validation. This is required to ensure that all terminology components added, updated or inactivated in the extension comply with all automatically verifiable authoring principles, including concept model rules, the SNOMED CT logical design and referential integrity constraints. After the authoring process, it is important to also perform a human review of all authored content to ensure that relevant editorial guidelines and principles are followed (including those which may not be able to be automatically checked), and that the updated content is acceptable from an author and user perspective. Finally, automated validation is required before the extension is released to ensure the correctness and consistency of the release as a whole.
The following pages explain each of these key validation steps further.

5.5.1 Validation During Authoring

Extension producers should ensure that validation is performed during the authoring process to ensure that new and updated content complies with editorial principles. An authoring tool should perform validation checks for all editorial rules that can be tested automatically. Additionally, classification during authoring is recommended, as it ensures that the content within the extension sits in the SNOMED CT hierarchy as expected and that no errors have been introduced. For more information on classification, please see 5.6.1.1 Classifying an Edition.

The table below provides some examples of automatic validation checks that should be performed when authoring components and reference sets. Please note that this table presents rules that are relevant for all SNOMED CT extensions, but there may be additional national linguistic and/or modelling guidelines which shoud be followed.

Table 5.5.1-1: Automatic validation of components and reference sets during authoring

Validation Type	Purpose	Examples
Components	To ensure that the components comply with authoring principles, at the time of authoring.	• Each concept must be the source of at least one relationship of type |Is a| • Each concept must have at least one FSN and at least one synonym • All attribute relationships specified for a given concept are valid to use in the given domain (based on the concept model rules defined in the MRCM)

Reference Sets	To check, at the time of authoring, that reference sets and their members are created and modified according to the associated specification and principles.	Common reference set checks
		• The reference set and its members comply with the reference set pattern specified in the |Reference set descriptor reference set|
		• Reference set attribute values of type |Component type|are available in either
		  • The extension,
		  • The International Edition, or
		  • A module from another extension on which the given extension depends.

		Map reference set checks
		• The map target must specify a valid code in the target code system
		• Every source code in a map that uses map rules must
		  • Have at least one group starting with mapGroup 1
		  • End with a "TRUE" rule

5.5.2 Post Authoring Review

Human review of all new and updated content in an extension is important to ensure the quality and correctness of the extension. This allows editorial rules that can not be automatically tested to be checked by an author who was not directly involved in the authoring of the given content. Post authoring review should be performed by individuals with both knowledge of SNOMED CT editorial principles as well as adequate clinical knowledge. In some situations, additional national linguistic or modelling guidelines also apply. A reviewer should, for example, be able to assess whether the definition of a concept has been authored correctly. The table below describes a range of different types of post authoring review, and provides some examples of each. For further information, please refer to the editorial guide.

Table 5.5.2-1: Post authoring review of components and reference sets in an extension

Review Type	Purpose	Examples

Components	To validate that the components created within the extension comply with editorial guidelines and are clinically correct.	• Review fully specified name (FSN) to ensure that it provides an unambiguous linguistic representation of the meaning of the concept.
		• Review all descriptions to ensure that they each follow editorial guidelines for Terming and Naming Conventions
		• Review the defining characteristics of new or updated concepts to ensure that:
		  • They each represent a true and necessary characteristic of the meaning of the concept
		  • If the concept is marked as fully defined, that the definition is sufficient to uniquely define the concept
		  • The concept complies with all relevant editorial principles and concept model rules

Reference Sets	**All reference set types** To validate that all reference sets meet their user requirements, such as the scope, size, functionality and user acceptance criteria. For more information, please refer to reference set review and quality assurance.	• Review the reference set to ensure that its quality is sufficient to meet the intended use cases of the reference set

	Subsets
	To validate that all members of the subset are within the intended scope of that subset, and that no component is missing from the subset that is required to meet the subset's intended purpose.
		• Review the set of *referencedComponentIds* to ensure that:
		  • All *referencedComponentIds* refer to a component that is in the intended scope of the subset.
		  • No component within the intended scope of the subset is missing from the set of *referencedComponentIds*.

	Maps
	To validate that the map between each SNOMED CT component and the associated codes from the other code system is correct
		• Review the set of maps to ensure that:
		  • Each mapGroup represents an appropriate set of map rules for the given *referencedComponentId*
		  • The mapTarget has a sufficiently similar meaning to the *referencedComponentId* to meet the user requirements of the map (given the associated *correlationId)*
		  • The mapRule is appropriate
		  • The mapAdvice is useful

Review Approaches

Collaborative authoring and review approaches are recommended to produce high quality content. Examples of authoring and review approaches include:

- **Single author with single reviewer**
  - One author develops the terminology content. Another author reviews the changes and either accepts them or reports issues that need to be considered and resolved.
- **Multiple authors with multiple reviewers**

- Two or more authors work on independent authoring tasks. Two or more reviewers then review the work of the authors. In most cases, the authors themselves act as the reviewers for the other authors' work.

- **Dual blind authoring with adjudicator**
  - Two authors work on the same task independently. For example, the authors may both map the same set of concepts to a target code system. Any discrepancy between the work of the two authors is automatically detected, An independent adjudicator then reviews the discrepancies and decides which author's work to approve.

5.5.3 Pre-release Validation

Prior to releasing a SNOMED CT extension or edition, a series of validation checks should be performed to ensure that the package is ready for release. Table 5.6.1.3-1 below summarizes the different types of validation tasks and provides some examples of checks that can be performed prior to a release.

Table 5.6.1.3-1: Pre-release validation tasks

Validation Type	Purpose	Examples
Structural Conformance	To validate that the SNOMED CT distribution files conform structurally to the specification of the associated file type, and that the rows and columns contain values of an appropriate data type.	• Concept file conforms to the structure specified in the Concept File Specification • Description file conforms to the structure specified in the Description File Specification • Relationship file conforms to the structure specified in the Relationship File Specification • Reference set files conform to the structure of the specific reference set type in Reference Set Release Files Specification
Release Type	To test assertions about the content of SNOMED CT data files, with respect to their release type (i.e. Full, Snapshot or Delta). This involves comparing data files for the prospective release with those of the most recently published previous release.	• New full file consists of the previously released full file and the new delta file • New delta file consists of all rows from the new full release which has an effectiveTime equal to the new release date • New snapshot file contains only one row for each component or reference set member

File	To test file constraints and interfile dependencies (i.e. assertions about the integrity of data within and between SNOMED CT data files). This involves testing primary and foreign keys, and the cardinality of references between files. For performance reasons, testing may be limited to the set of concepts that have changed in some way in the prospective release, and components associated with them.	• Active extension descriptions are referred to in one or more language reference sets
		• All concepts associated with an active description in the description file are present in the concept file of either the extension, the International Edition, or a module from another extension on which the extension modules depend.
		• Primary keys are unique within each file
		  • For files in the full release, the combination of the *id* and the *effectiveTime* is unique
		  • For files in the snapshot and delta releases, the *id* is unique

Component	To test assertions about the integrity of SNOMED CT components. In this kind of validation, the content of each file is tested against the editorial principles and logical design of SNOMED CT.	• All active concepts must have:
		  • An active description of type |Fully specified name|
		  • At least one active description of type |Synonym|
		  • At least one active |is a| relationship
		  • A transitive |Is a| relationship to the root concept
		  • A *definitionStatusId* which refers to an active descendant of |Definition status|
		• All active relationships must conform to the following rules:
		  • *sourceId* and *destinationId* must both refer to active concepts
		  • *typeId* must refer to an active descendant of |Attribute|
		  • *characteristicTypeId* and *modifierId* must refer appropriate active metadata concepts
		• All active descriptions must conform to the following rules:
		  • *conceptId* must refer to a valid concept (which may be active or inactive)
		  • *typeId* and *caseSignificanceId* must refer to appropriate active metadata concepts

5.6 Distribution

Distribution is the process of making the extension available to its consumers.

It is the responsibility of the extension provider to ensure that everyone who accesses the SNOMED CT distribution has an appropriate license to do so. The extension producer must issue Affiliates with a license to use the extension. These licenses must include a requirement for the licensee to hold a SNOMED CT Affiliate License for the International Release. For more information about Members' responsibilities in relation to licensing, please see 5. The role of NRCs related to SNOMED CT licensing.

The following pages will examine the steps involved in preparing an extension for distribution and introduce different approaches to distribution.

5.6.1 Preparing for Distribution

Before an extension is ready for distribution it needs to be classified (if distributed as an edition), validated and packaged properly. Extension producers also need to ensure their content is aligned with the International Edition of SNOMED CT. The following pages explore several key aspects of preparing a SNOMED CT edition for distribution, including:

5.6.1.1 Classifying an Edition

When new concepts are added to an extension, or the definitions of existing concepts are modified, it is important that the extension is classified. Classification is performed for two main reason: firstly to ensure that logic errors are identified, and secondly to make it easy for users of the terminology (who may not necessarily have access to a classifier) to identify the full set of inferred relationships.

Classifying an extension requires combining the content in the extension modules with the content from all modules on which the extension modules depend (as defined by the Module Dependency Reference Set), including the modules from the International Edition. This ensures that the logical definition of all supertypes and attribute values of extension concepts can be used by the classifier in determining its inferences. The resulting set of inferred relationships (excluding redundant |is a| relationships) is then distributed in the Relationship file of the release.

There are, however, some limited situations in which classification may not be required in order to generate the (inferred) Relationship file for an extension. For example:

- The extension contains only reference sets, metadata concepts and/or descriptions
- The extension contains only primitive concepts for which the stated and inferred definition are equivalent
    - *Note: Some primitive concepts may infer new defining relationships during the classification process, so care should taken before assuming that classification is unnecessary for primitive content.*

Purpose of Classification

The use of description logic as the formal foundation of SNOMED CT allows the semantics of clinical concepts to be represented unambiguously. Description logic also enables logical deduction in which additional information can be inferred from the explicit statements in the terminology. Classification is the process in which the formally stated definitions of each concept are used to compute the subsumption hierarchies and defining properties of each concept.

With every release of SNOMED CT, two types of relationship files are distributed: the Stated Relationship file, containing the defining relationships stated by the terminology authors; and the Relationship file, containing the full set of relationships that can be inferred using a classifier (excluding non-redundant |is a| relationships). It is important that extension producers are aware of the difference between these two files, and the fact that most consumers of the terminology will use the Relationship file (containing the inferred relationships). The Stated Relationship file is primarily used by extension producers, but may also be used by terminology consumers that have access to a description logic classifier.

Stated Relationships

The Stated Relationship File contains the stated form of SNOMED CT. The stated form of a Concept is the Description Logic definition that is directly edited by authors or editors. It consists of the stated 116680003 |is a| relationships plus the defining relationships that exist prior to running a classifier on the logic definitions. Therefore, the stated form of a Concept is represented by a collection of relationships: one or more 116680003 |Is a| relationships and zero or more defining relationships.

The Stated Relationship File is in the same table format as the Relationship File, but the value of the characteristicTypeId field is 900000000000010007 |Stated relationship (core metadata concept)|.

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

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

Inferred Relationships

Inferred relationships are derived from the set of stated defining relationships, by applying a consistent set of logical rules to the definition which take account of the definitions of related concepts.

Several semantically equivalent views may be inferred from the same set of stated relationships. However, the SNOMED CT Relationship file is distributed using an inferred view known as the Distribution Normal Form (DNF). This standard distribution view, includes all non-redundant |is a| relationships (between each concept and its proximal supertype), and the inferred concept definition of each concept (including all non-redundant defining relationships). For more information, please refer to the Inferred definition views section in the Terminology Services Guide.

Please note that in the standard distribution view (DNF):

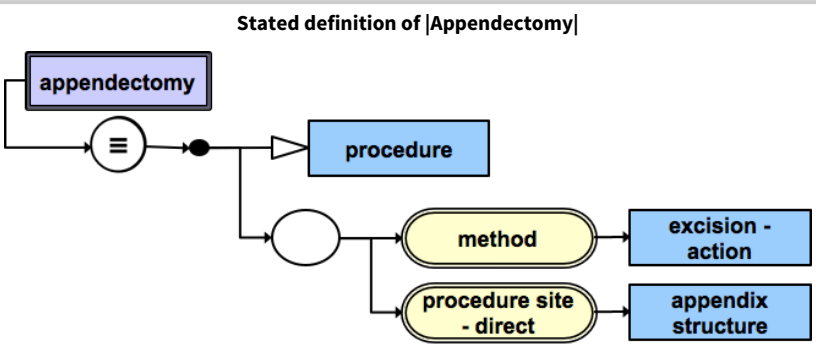
- Redundant |Is a| relationships are removed
- Defining attribute relationships are inherited from the concept's supertype parent, unless the attribute is also stated with a more specific value

Example

This example is included to illustrate what happens during the classification process.

Consider the stated relationships shown below for the concepts |Appendectomy| and |Emergency appendectomy|. The concept definitions are represented in accordance with SNOMED CT Diagramming Guidelines.

Table 5.6.1.1-1: Example stated relationships

This diagram shows the stated definition of the concept |Appendectomy|. An |Appendectomy| is a procedure in which the appendix structure is excised.	

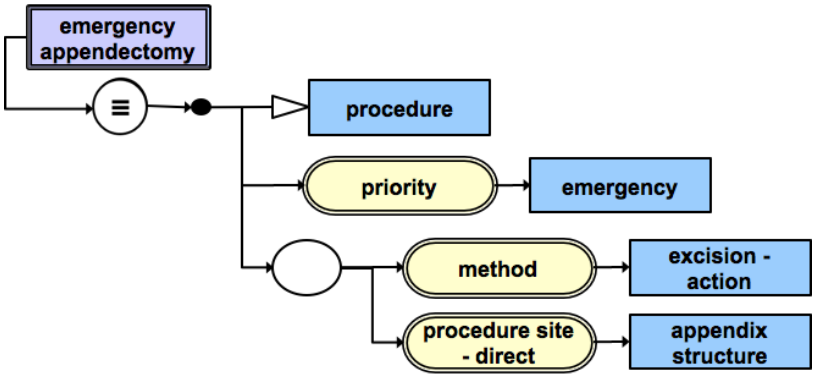
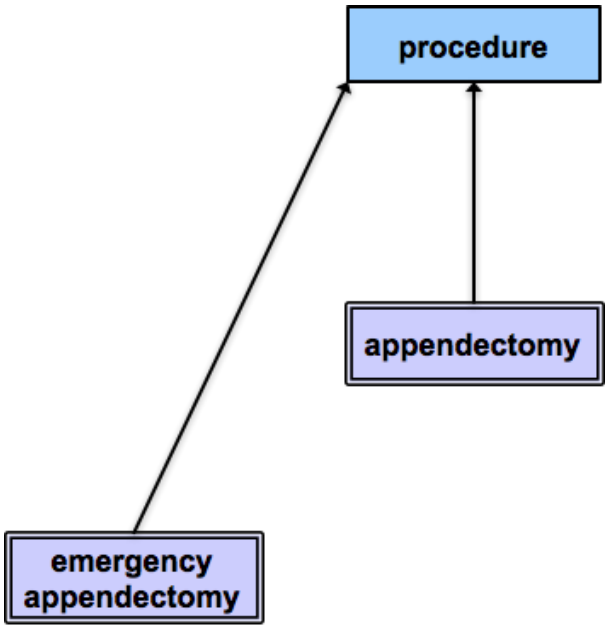
This diagram shows the stated definition of the concept |Emergency appendectomy|. An |Emergency appendectomy| is a procedure with a priority of emergency, in which the appendix structure is excised.	**Stated definition of |Emergency appendectomy|** 
This diagram shows a hierarchical view of the stated subtype relationships from the concepts above.	**Stated subtype relationships** 

Table 5.6.1.1-2: Comparison of stated definitions

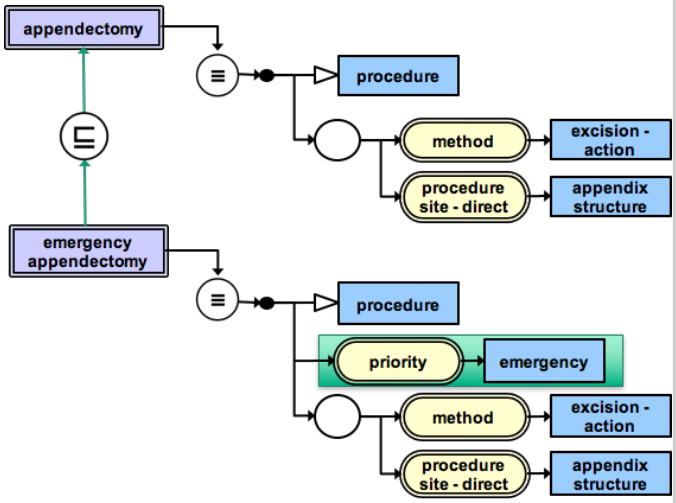
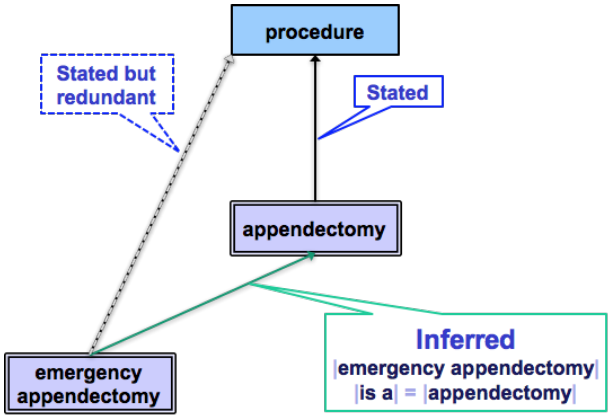
This diagram shows a comparison of the stated definitions for |Appendectomy| and |Emergency appendectomy| from above. The two definitions are identical, except for the additional defining relationship on |Emergency appendectomy|, which states that the priority is emergency. This means that a classifier can infer that |Emergency appendectomy| is a logical subtype of |Appendectomy|.	

Table 5.6.1.1-3: Deriving the inferred relatonships

This diagram shows that once the relationship |Emergency appendectomy| |is a| |Appendectomy| is inferred, the stated relationship |Emergency appendectomy| |is a| |Procedure| becomes redundant.	

Combining Modules for Classification

Classifying an extension requires combining the content in the extension modules with the content from all modules on which the extension modules depend (as defined by the Module Dependency Reference Set), including the modules from the International Edition. This ensures that the logical definition of all supertypes and attribute values of extension concepts can be used by the classifier in determining its inferences. The resulting set of inferred

The process of combining an extension with the content of the modules on which it depends is illustrated in Figure 5.6.1.1-1 below.

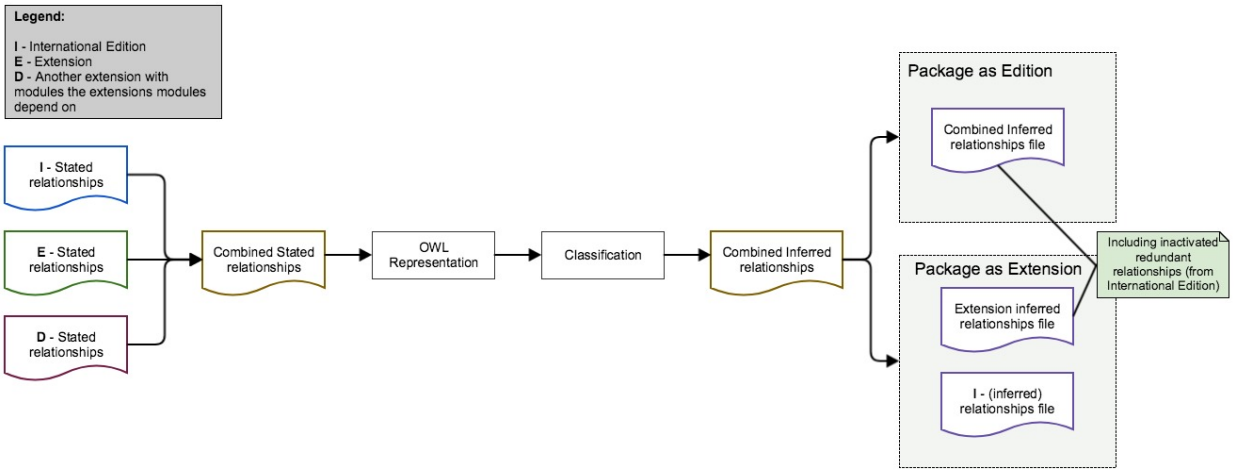


Figure 5.6.1.1-1: Creating inferred relationships for an extension

It is important that all inferred relationships present in the International Edition are also present in the combined inferred view. This means that the Relationship file in an extension's edition should be a superset of the Relationship file in the International Edition. In particular:

- All relationships belonging to the International Edition should be retained in the extension edition, with the same *moduleId* and *effectivetime* values
- All new inferred relationships created when classifying the extension should
    - Be assigned to a module within the extension, and
    - Use an *effectiveTime* which corresponds to the release date of the extension

If a situation occurs in which an inferred relationship from the International Edition becomes redundant in the extension edition (due to an intermediate concept being created in the extension), it may be necessary to inactivate the redundant international relationship in an extension module. For more information please refer to 5.4.4.3 Inactivate Relationship in an Extension.

For more information about options for packaging inferred extension relationships, please refer to 5.6.1.2 Packaging and File Naming.

5.6.1.2 Packaging and File Naming

Purpose

The purpose of the SNOMED CT packaging process is to assemble a set of SNOMED CT files into an archive of a specified structure to support easy adoption by terminology consumers. The resulting archive is a distributable package for the specific extension or edition. A SNOMED CT package uses a standard naming convention and its contents are structured in a standard format. The packaging process may begin when all the files that comprise a specific SNOMED CT extension or edition have been created, populated and validated.

Additional Packaging Notes:

1. Each SNOMED CT release package may logically consist of one or more editions, with each edition consisting of the set of components and reference set members which belong to a focus module, plus the contents of the modules on which the focus module depends.
2. The module dependencies used to define the contents of an edition are represented using a Module Dependency Reference Set.
3. SNOMED CT editions can be identified using a URI that is formatted according to the SNOMED CT URI Standard. (For more specific information please refer to URIs for Editions and Versions.)

Release File Packaging

SNOMED CT content in the International Edition is organized into three top level folders according to the release file types - Full, Snapshot and Delta. When packaging extensions and editions, this should be done using the same folder structure. Note that extension producers are only required to include the full release type for the extension. The delta and snapshot releases are optional, because they can be calculated from the full release. However, distribution of all three release file types is recommended to simplify use by terminology consumers.

Each release file type has the same nested folder structure, with the following two subfolders:

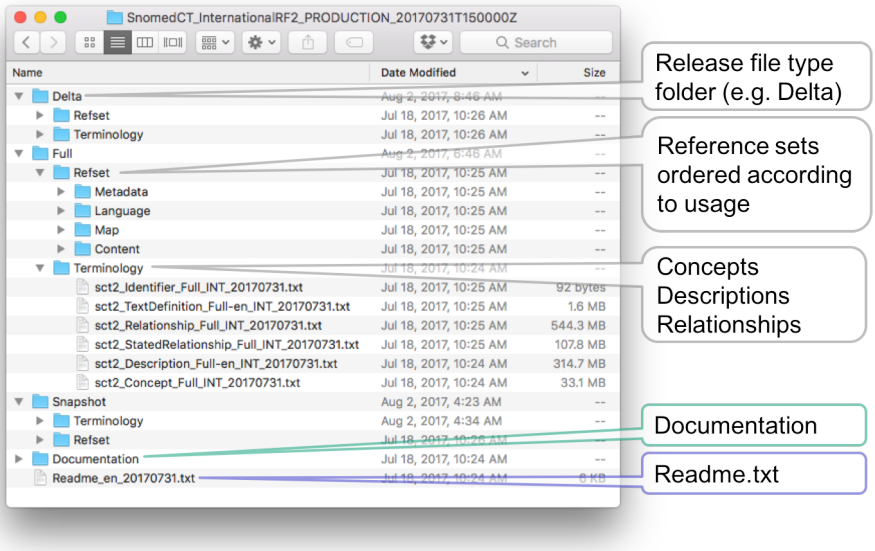
- **Terminology** folder, which contains the concepts, descriptions and relationships files
- **Refset** folder, which contains a range of reference set files ordered into subfolders according to their usage

The following recommendations apply to the structure and format of SNOME CT release packages:

1. It is recommended that documentation should be removed from the release packages, and hosted separately. This allows for ongoing updates, if required.
    a. Note that some exceptions may apply, such as the readme.txt file
2. The main component files (e.g. Concept, Description, Relationship) are nested under the Terminology folder
3. All reference sets files are nested under the Refset folder, in the relevant subfolder:
    a. **Content** folder contains reference sets that represent SNOMED CT subsets
    b. **Language** folder contains a language reference set, for each applicable language or dialect
    c. **Map** folder contains simple, complex and extended map reference sets
    d. **Metadata** folder contains supporting metadata files, including the reference set descriptor reference set, the module dependency reference set, and the machine readable concept model (MRCM) reference sets

The recommended folder structure is illustrated below.

General Release Folder Structure	Example - International Release

• Full
  • Terminology
  • Refset
  • Content
  • Language
  • Map
  • Metadata
• Delta
  • Terminology
  • Refset
  • Content
  • Language
  • Map
  • Metadata
• Snapshot
  • Terminology
  • Refset
  • Content
  • Language
  • Map
  • Metadata
	

SNOMED International provides templates to support extension producers in proper packaging of the release files. These templates detail the minimum expected set of files for each release product, plus the folder structure in which they should be packaged. For more information and to download these templates, please refer to https://confluence.ihtsdotools.org/display/RMT/Release+Package+Templates

Packaging Options

Two main approaches to packaging exist:

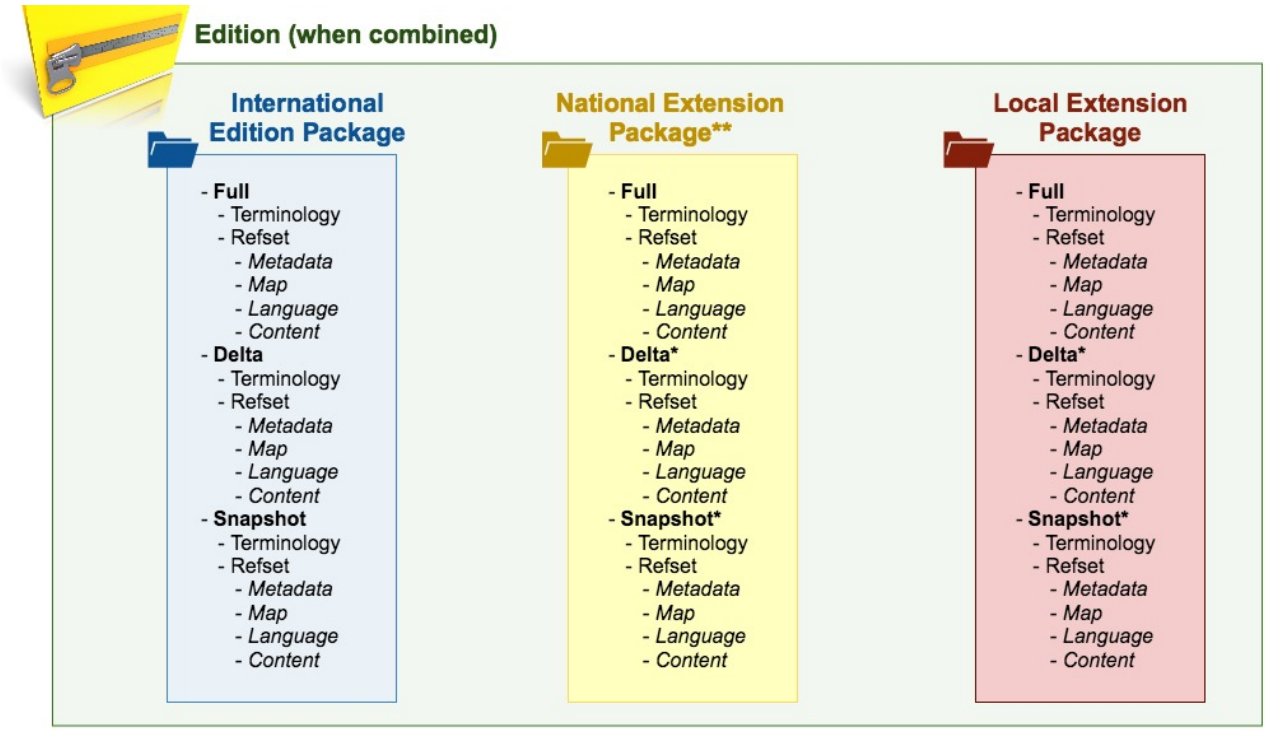
- **Extension:** Packaging the extension modules separately from the modules in other editions or extensions
- **Edition:** Packaging the extension modules together with the modules from the International Release (and other modules on which the extension depends), as a single package with the terminology content available in combined release files

The approach that is best for a given situation will depend on the content of the extension and the requirements of the consumers of the extension. The following subsections discuss each approach.

Packaging as an Extension

Figure 5.6.1.2-3 below illustrates the idea of packaging as an extension. In this approach, the extension content is released in a separate package, which is not intended to be used on its own. Instead, the contents of the extension package must be combined with other packages (including the international release) by the terminology consumer. To determine which packages must be combined, the module dependency reference set is used to identify the dependencies for a given SNOMED CT edition (based on its focus module).

In the example below, the national content and local content have each been packaged as extensions. A terminology consumer must therefore combine the Local Extension package, the National Extension package and the International Edition package to achieve a complete terminology solution. Note that each package uses the same folder structure.



** These folders are recommended but not required for extension releases*
*** May be required if the modules in the local extension are dependent on modules in the national extension*

Figure 5.6.1.2-1: Creating a SNOMED CT Edition from extension packages

Extension producers choosing this packaging approach may package the content of the extension into release files in a variety of ways, including:

- All content for a particular type of component (e.g. Concept) that is maintained by the extension producer is released in a single file. Components in this file may have different moduleIds, where the content has been authored in separate modules. Please note that where descriptions are authored in more than one language, these are generally included in separate files, with the applicable language code included in the file name.
- All content for a particular type of component (e.g. of type Concept) is included in a set of files, with each file using a single moduleId.

Extension packaging may be appropriate when the content has the following characteristics:

- The extension is used only to distribute a translated version of the terminology
- The extension is used only to distribute reference sets and the associated metadata, and does not involve the addition of clinical concepts
- The modules in the extension are intended to be reused by multiple editions, which will each be classified with a different combined set of modules

Packaging as an Edition

Figure 5.6.1.2-2 below illustrates the idea of packaging as an edition. In this approach, the package can be used on its own, without the need to combine with other packages.

When packaging an extension as an edition, the extension content is combined with the International Edition (and any other module on which the extension depends), in the standard folder structure. All content of a particular type is included in a single file, irrespective of the module it belongs to or the organization responsible for maintaining it. Care should be taken not to modify, add to or remove content that belongs to a module maintained by another organization. For more information please refer to authoring extensions.

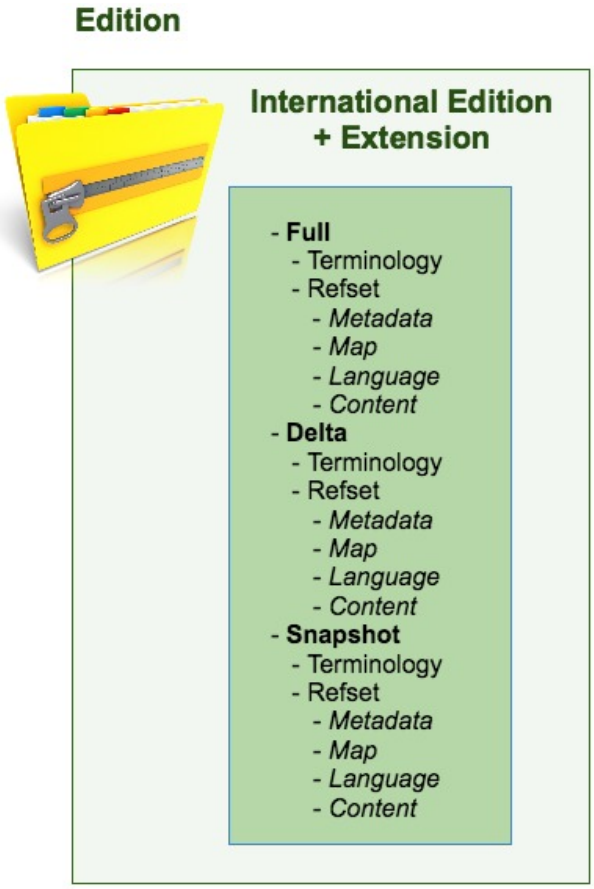


Figure 5.6.1.2-2: Packaging an extension as an edition

Edition packaging may be appropriate when the content has the following characteristics:

- The extension is used only to distribute a translated version of the terminology
- The extension is used only to distribute reference sets and the associated metadata, and does not involve the addition of clinical concepts
- The extension includes new clinical concepts that need to be classified together with the International Edition (and other modules on which the extension depends). See 5.6.1.1 Classifying an Edition.

File Naming

Files in an extension should be named in accordance with the SNOMED CT file naming convention. The file naming convention can simplify implementation and provide the following benefits:

- A consistent naming convention across the International Edition and each National Edition
- Predictable file naming which provides a stable pattern for naming over time and between releases
- A standard way to identify the source and namespace by which a release file is managed
- A consistent versioning mechanism
- A mechanism to identify the contents of a file at a high level
- A mechanism to identify the type of information stored in a release file (e.g. documentation, tooling, etc.)

- Guidance on file naming for release files in non-English extensions
- Assurance that names will be unique across all editions and extensions over time

Quality assurance checks, which ensure that the naming convention has been applied, should be performed as part of the release process. For more information please refer to 2.1.2. Release File Naming Convention.

5.6.1.3 Release Validation

Prior to releasing a SNOMED CT extension or edition, a series of validation checks should be performed to ensure that the package is ready for release. Table 5.6.1.3-1 below summarizes the different types of validation tasks and provides some examples of checks that can be performed prior to a release.

Table 5.6.1.3-1: Pre-release validation tasks

Validation Type	Purpose	Examples
Structural Conformance	To validate that the SNOMED CT distribution files conform structurally to the specification of the associated file type, and that the rows and columns contain values of an appropriate data type.	• Concept file conforms to the structure specified in the Concept File Specification • Description file conforms to the structure specified in the Description File Specification • Relationship file conforms to the structure specified in the Relationship File Specification • Reference set files conform to the structure of the specific reference set type in Reference Set Release Files Specification
Release Type	To test assertions about the content of SNOMED CT data files, with respect to their release type (i.e. Full, Snapshot or Delta). This involves comparing data files for the prospective release with those of the most recently published previous release.	• New full file consists of the previously released full file and the new delta file • New delta file consists of all rows from the new full release which has an effectiveTime equal to the new release date • New snapshot file contains only one row for each component or reference set member

File	To test file constraints and interfile dependencies (i.e. assertions about the integrity of data within and between SNOMED CT data files). This involves testing primary and foreign keys, and the cardinality of references between files. For performance reasons, testing may be limited to the set of concepts that have changed in some way in the prospective release, and components associated with them.	• Active extension descriptions are referred to in one or more language reference sets
		• All concepts associated with an active description in the description file are present in the concept file of either the extension, the International Edition, or a module from another extension on which the extension modules depend.
		• Primary keys are unique within each file
		  • For files in the full release, the combination of the *id* and the *effectiveTime* is unique
		  • For files in the snapshot and delta releases, the *id* is unique

Component	To test assertions about the integrity of SNOMED CT components. In this kind of validation, the content of each file is tested against the editorial principles and logical design of SNOMED CT.	• All active concepts must have:
		  • An active description of type |Fully specified name|
		  • At least one active description of type |Synonym|
		  • At least one active |is a| relationship
		  • A transitive |Is a| relationship to the root concept
		  • A *definitionStatusId* which refers to an active descendant of |Definition status|
		• All active relationships must conform to the following rules:
		  • *sourceId* and *destinationId* must both refer to active concepts
		  • *typeId* must refer to an active descendant of |Attribute|
		  • *characteristicTypeId* and *modifierId* must refer appropriate active metadata concepts
		• All active descriptions must conform to the following rules:
		  • *conceptId* must refer to a valid concept (which may be active or inactive)
		  • *typeId* and *caseSignificanceId* must refer to appropriate active metadata concepts

5.6.1.4 Preproduction Releases

Prior to the release of a SNOMED CT Edition it is strongly recommended that an alpha and/or beta version of the edition is created and released. This will give stakeholders an opportunity to test the edition, gain an understanding of any impact to their local systems, and identify potential errors that need correction.

Alpha Release Package

A SNOMED CT release package that is only being released for initial review and testing by implementers and other stakeholders and must not be used in production clinical systems or in clinical settings.

Notes

1. The objective of an *alpha release* is to test the chosen approach and elicit feedback before committing to the content and/or release format for the additional material. It is likely that, prior to publication of a beta release, significant changes will be made to address the feedback received, and issues identified by testing.
2. *Alpha releases* should not be distributed to Affiliate Licensees or any third parties except those who have formally committed to take part in an approved evaluation process.
3. *Alpha releases* must not be used in an operational environment that may incorporate the data into a record or create a dependency on continued maintenance of the additional components or derivatives.
4. *Alpha releases* were formerly known as a “Technology Preview” releases.

Beta Release Package

A SNOMED CT release package that is only being released for review and testing by implementers and other stakeholders, prior to release of a production release, and must not be used in production clinical systems or in clinical settings.

Notes

1. The *beta release* status indicates the releasing organization expects to subsequently confirm it as a production release. However, if a significant issue is reported in its format or content during the feedback period, the releasing party reserves the right to withdraw a *beta release*, or to replace it with an updated *beta release package*. Therefore, the releasing organization does not commit deciding whether this will be treated as a production release until shortly before the due date for the next release. If a *beta release* is subsequently be confirmed as a production release, all subsequent updates to the additional components and derivatives will be fully version tracked from date of that *beta release*.
2. Anyone testing the use of a *beta release* must be prepared for withdrawal or significant changes that may occur to the additional components or derivatives. Therefore, this data should not be used in an operational environment in ways that create a dependency on continued maintenance of the additional components or derivatives.
3. Beta releases should not be distributed to Affiliate Licensees or any third parties except those who have formally committed to take part in an approved evaluation process.
4. Beta releases were formerly known as “Candidate Baseline” releases.

A beta release of an extension functions as a candidate baseline until the extension is considered mature and ready for publication. The duration of a preproduction release period can be as long as is required to ensure that the version that is finally published is as robust and error-free as possible. Extension producers should utilize the option of distributing preproduction releases to get feedback from relevant stakeholders on the content of the extension. This is particularly important because SNOMED CT's robust versioning mechanism means that any content that is officially published by an extension producer must be maintained indefinitely. Errors identified and corrected in an alpha or beta release can be fixed without requiring an ongoing history to be maintained. In contrast, errors identified in a published release must be traceable and made permanently available in the full history of the edition.

5.6.2 Distribution Approaches

There are two main approaches to distributing SNOMED CT extensions and editions.

- File-based distribution
- Service-based distribution

Either or both of these approaches may be applied by extension producers to make their content available to consumers.

File-Based Distribution

The most common approach to distribution is using a file-based distribution approach, in which the SNOMED CT content is accessed via a distribution platform that allows a complete release package to be downloaded, as illustrated in the diagram below.

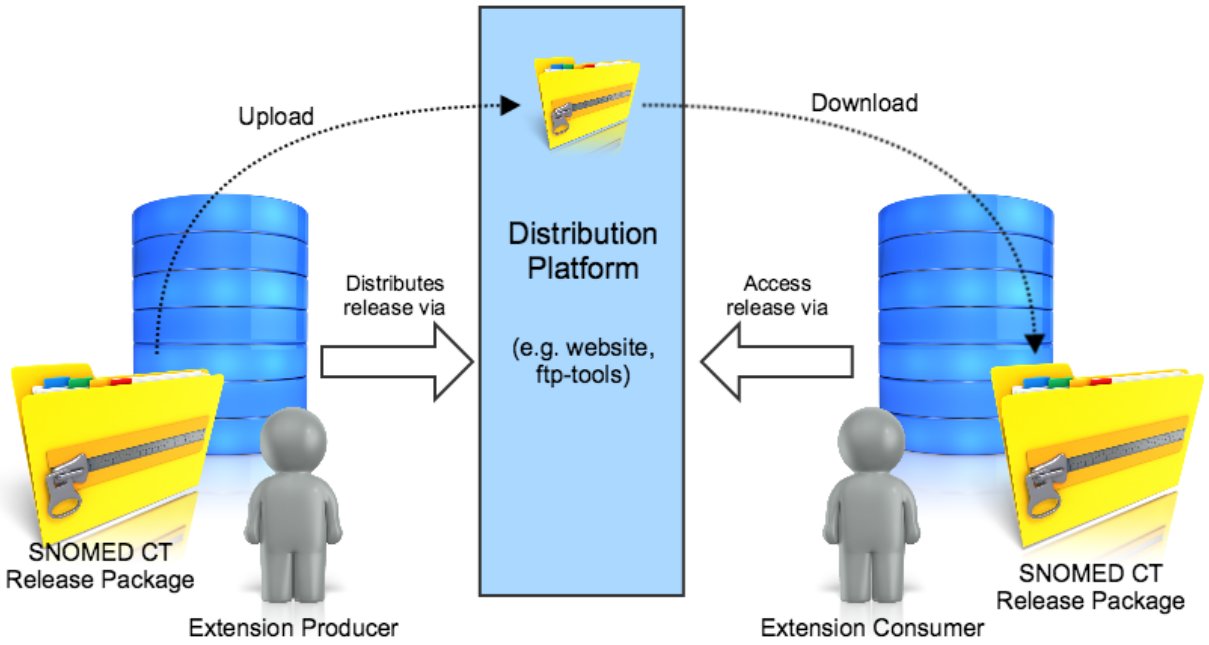


Figure 5.6.2-1: File-based distribution

Examples

Member Licensing and Distribution Service

SNOMED International makes the International release available as downloadable packages, using a file-based distribution approach. The Member Licensing and Distribution Service (MLDS) is an online service for managing affiliate licenses for SNOMED International and participating Members, and for distributing SNOMED International products to affiliate licensees. An increasing number of SNOMED International Members use the MLDS to distribute their National Edition, including Denmark, Sweden, Norway, Belgium and the Netherlands. The MLDS serves two types of users:

- **SNOMED International and NRC staff** – Staff of organizations, including SNOMED International, who are using the MLDS to distribute SNOMED CT editions and extensions.
- **Affiliates and interested parties** – Individuals and organizations who agree to a license to download and use SNOMED CT in non-Member countries, and in Member countries who are using the MLDS to distribute their national extension. Please note, that when using SNOMED CT in non-Member countries, an Affiliate license is required, and the use is subject to fees. For more information, please refer to SNOMED CT Licensing.

The MLDS service can be accessed at MLDS.

Plaftorms Provided by a National Release Centre

In some Member countries, SNOMED CT is downloaded from a local platform provided by the National Release Center (NRC) of that country. For example:

- In the US - National Library of Medicine / National Institutes of Health
- In Canada - Canada Health Infoway
- And in the UK - NHS Digital's UK Terminology Centre

To download SNOMED CT in a Member country:

- Visit to the SNOMED International Members page at http://snomed.org/members and find your member organization
- Visit your NRC's website or contact them by email
- Download the relevant release files from your NRC
    - Note that most NRCs will require you to register for an account before SNOMED CT can be downloaded
    - In some countries SNOMED CT is alternatively provided via terminology services

Service-Based Distribution

Another option for distributing SNOMED CT is using a service-based approach. This approach allows users to access the entire release, or specific parts of the release via query services using a set of predefined Application Programming Interfaces (APIs).[1] Service-based distrbution is illustrated in the diagram below.

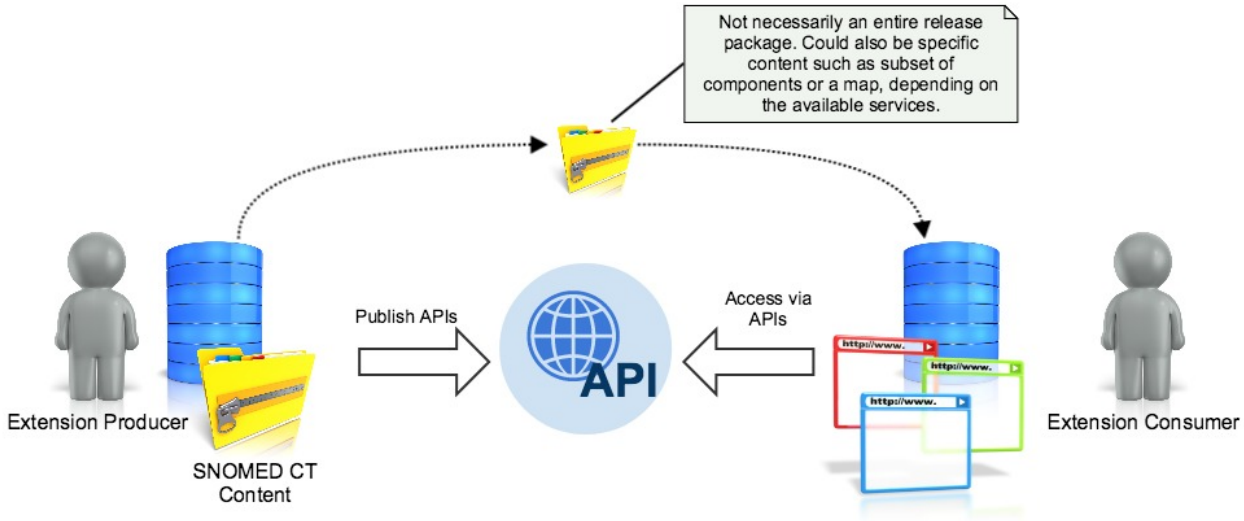


Figure 5.6.2-2: Service-based distribution

The extension producer may choose to provide different types of service-based distribution services, such as:

- A service that provides access to specific SNOMED CT subsets
- A service that provides access to the content within a specific set of modules (e.g. the extension modules)
- A service that provides access to the content in an entire edition, including both the extension content and the modules on which the extension depends

Using a terminology server API, extension producers can support the needs of extension consumers, without the consumer needing to download the full release files themselves. For example, the service could allow access to specific reference sets, support querying for the descendants of a particular concept, or determine the permitted range for a given attribute. Distribution services may also provide options for accessing different editions, and different versions of those editions.

As the range of possible terminology services is extensive, extension producers need to decide on an appropriate API to meet the needs of their consumers. Options include both standardized and commercial APIs, such as:

- SNOMED International's SNOMED CT Snapshot API
- HL7 FHIR terminology services API
- HL7 Common Terminology Services 2 (CTS 2)
- Commercial terminology server APIs that offer SNOMED CT search and query

[1] An Application Programming Interface is a set of rules and specifications that enable communication between software programs. *Application Programming Interfaces* enables interaction between separate software programs, in much the same way that a user interface facilitates interaction between humans and computers.

5.6.3 Implementation Guidance

Extension producers should ensure that the consumers of their extension understand how to use and implement the extension. It is therefore recommended that extension producers provide specific guidance on the:

- Purpose of the extension
- Scope and content of the extension
- Module dependencies within the extension
- Usage of the extension

Additionally, it is important that consumers of the extension can trust the quality of the extension and understand how to validate its contents. It is therefore also recommended that guidance on validating the extension is provided.

> Please note that this guide focuses on guidance for extension producers. Specific implementation guidance for extension consumers should be provided by the extension producer themselves, to supplement the general implementation guidance provided by SNOMED International (see http://snomed.org/doc).

5.7 Maintenance

Maintenance of an extension is essential to ensure that the extension continues to meet emerging user requirements, and remains aligned with updates to the International Edition of SNOMED CT (and any other extension on which it depends). As illustrated in Figure 5.7-1 below, the extension maintenance period begins when a new version of the extension is published and ends when the next version of the extension is published. As such, there is a continuous cycle of maintenance.

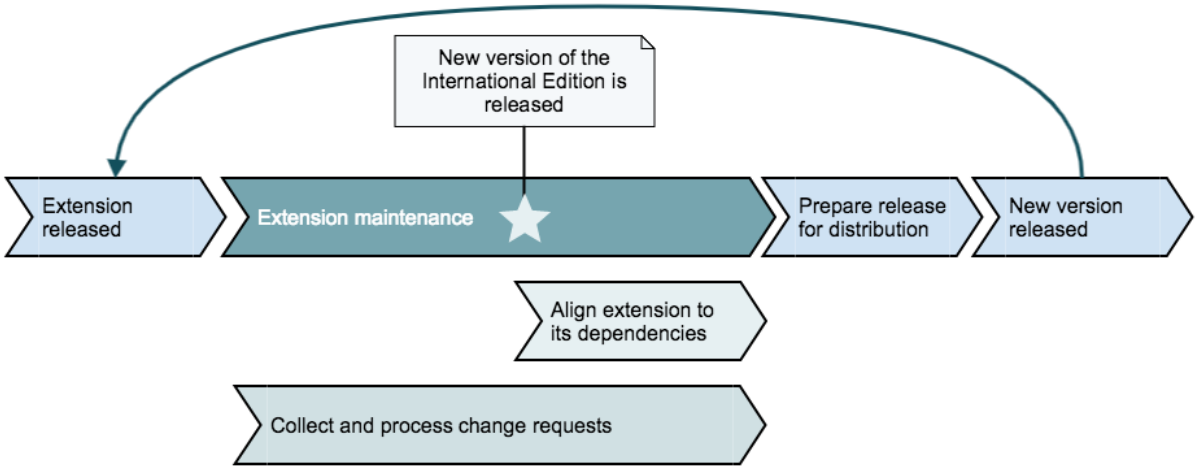


Figure 5.7-1: Extension maintenance cycle

Two key maintenance tasks should be performed by an extension producer:

- Collecting and processing changes requested by terminology consumers
- Aligning the extension to new versions of the International Edition or other extensions on which it depends

These two maintenance tasks are explained further in 5.7.2 Change Requests and 5.7.1 Change Management respectively.

Extension Release Cycle

Regular updates to SNOMED CT are essential to satisfy emerging user requirements and to improve the quality of SNOMED CT itself. An extension producer must be aware that changes in the International Edition (or another extension on which the producer's extension depends) may have an impact on their extension. Therefore, the extension should be reconciled with the International Edition (and other associated extensions) every time a new release becomes available.

The International Edition of SNOMED CT is currently released twice a year - on January 31st and July 31st. Prior to each release of the International Edition, National Release Centres have an opportunity to evaluate the pre-production releases and plan their extension reconciliation strategy. Once the International Edition has been officially released, reconciliation of the extensions based on the new release can begin. Once complete, it is recommended that extension producers allow plenty of time for an alpha and/or beta version of their extension to be validated by their users before the new release is officially published. See 5.6.1.4 Preproduction Releases for more information.

5.7.1 Change Management

Each new release of the SNOMED CT International Edition introduces some changes to the content in SNOMED CT. Many of these changes are additions to the breadth and depth of clinical coverage. Other changes may include new descriptions, corrections to concept definitions, and enhancements to the expressivity of the concept model. These changes are an essential characteristic of an evolving clinical terminology which seeks to support current requirements. However, each change made to the International Edition may have an impact on the content in an extension. It is therefore important to evaluate and manage these changes when reconciling an extension with a new version of the SNOMED CT International Edition (or another extension on which it depends).

The change management process can be divided into three distinct phases:

1. Identify changes
2. Assess change impact

3. Update extension

After change management has been performed, it is important that thorough review and validation of the resulting extension is performed.

In the following sections we discuss the three phases of change management.

5.7.1.1 Identifying Changes

Given the robust versioning mechanism of SNOMED CT and the machine processable distribution format, changes made in a new version of a SNOMED CT edition are easy to identify automatically. Extension producers should ensure that they have appropriate mechanisms in place to identify changes in any edition on which their extension is dependent. To identify changes in an edition, the following release artifacts can be used:

- The delta files from the new release
- The snapshot files from the previous release

These two release artifacts enable changes to be identified, and the nature of those changes understood. The delta release files provide information about all components or reference set members which have changed since the previous release. By comparing this information to the corresponding components or reference set members in the snapshot files from the previous release, stakeholders can understand which of these changes represent additions of new components or reference set members, and which represent new versions of existing components or reference set members. For more information on the different release types please refer to the release file specification.

The table below shows how the values for the *active* attribute in the previous snapshot and current delta can be compared to help to determine the types of change in the release.

		Row in the DELTA view of the CURRENT release		
		active = 0	**active = 1**	**row not present**
Row in the SNAPSHOT view of the PREVIOUS release	**active = 0**	-	Component / member REACTIVATED	NO CHANGE
	active = 1	Component / member INACTIVATED	Component / member CHANGED	NO CHANGE
	row not present	-	Component / member ADDED	-

The types of changes that components and reference set members can undergo are inactivate, reactivate, change or add. For example, if a row (with a given *id*) was not present in the snapshot from the previous release, and then was present (with the same *id*) in the current delta release with an active value of 1, then this means that a new component or reference set member has been added. If a row (with a given id) had a value of 1 in the previous snapshot, and now has a value of 1 in the current delta, then this means that the component or reference set member has been changed.

⚠ Please note, SNOMED International strongly recommends that extension producers perform regular alignment to the International Edition, without skipping any version. If, for some reason, an extension is not updated to align with a new version of the International Edition, additional maintenance tasks will be required. For example, the extension producer will need to generate a modified 'delta view', which contains all changes made since the last version of the International Edition that was reconciled with the extension. All components and reference set members with an effectiveTime that is more recent than the last reconciled version of the International Edition will need to be considered in this 'delta view'.

5.7.1.2 Assessing Change Impact

After identifying the components that have changed since the previous release, it is important to assess how these changes may affect the content of the extension. In particular, extension producers will need to ensure that the quality of the extension has not been compromised. For example, this may include checking that no relationship in the extension refers to an International Edition concept that has been inactivated, and checking for new concepts that may need to be added to reference sets in the extension.

The following table presents some examples of how changes in the International Edition may affect new versions of an extension. Please note that this list is not exhaustive, but instead provides examples of the types of maintenance tasks which are likely to be required when managing changes in an extension. Also note that extension consumers will need to assess the impact of changes in an extension edition on their implementations.

Table 5.7.1.2-1: Examples of terminology changes that may affect an extension

Concept inactivations	Extension Impact
A concept, which is referenced by a reference set member (i.e. the *referencedComponentId*) in the extension, has been inactivated.	Reference sets are permitted to reference inactive components as there are valid use cases for this. However, it is recommended that inactive components are not for capturing clinical data. Therefore, if the extension reference set may be used by consumers for clinical data capture, the associated reference set member should be inactivated, and a potential replacement considered. For more information please refer to 6.3.2 Authoring Reference Sets. Potential replacements for the inactivated concept can be found in the relevant historical association reference set.
A concept, which is the parent of an extension concept (i.e. it is the *destinationId* of a relationship of type |is a|), has been inactivated.	To retain the referential integrity of the extension, it is important that all extension concepts are subsumed (via active concepts) by the root concept in the International Edition. When the parent of an extension concept is inactivated, the associated |is a| relationship in the extension must also be inactivated, and a new relationship created to link the concept to an alternative international parent concept. For more information please refer to 5.4.4.3 Inactivate Relationship in an Extension and 5.4.4.1 Add Relationship in an Extension.
A concept, which is used in an extension relationship (i.e. in the *sourceId*, *typeId* or *destinationId* of the relationship), has been inactivated.	When a concept is inactivated, all relationships in which that concept participates in (as the *sourceId*, *typeId* or *destinationId*) must also be inactivated. For more information please refer to 5.4.4.3 Inactivate Relationship in an Extension.
Concept additions	**Extension Impact**
A concept has been added to the International Edition, on request by the extension producer, and a corresponding concept has been temporarily added to the extension in the interim.	If a new concept is added to the International Edition, which represents the same meaning as a concept in the extension, the extension concept should be inactivated. Any references to the inactivated extension concept should be replaced by references to the new international concept. For more information, please refer to 5.4.2.3 Inactivate Concept in an Extension, 5.4.4 Authoring Relationships and 6.3.2 Authoring Reference Sets. In general, checks should to performed ensure that no fully defined concept in the extension has the same logical definition as a concept in the International Edition (or any module on which the extension depends).
A concept has been added to the International Edition, on request by the extension producer, for use in a national reference set.	A new member of the extension reference set should be added to refer to the new international concept. For more information, please refer to in 5.4.6.1 Add Members to a Reference Set.

Description inactivations	Extension Impact
A description, which is referenced by a reference set member (i.e. the *referencedComponentId*) in the extension, has been inactivated. For example, the description may be used in a national language reference set.	The reference set member that references the inactive description should be inactivated. For more information, please refer to 5.4.3.3 Inactivate Description in an Extension. In some situations, the inactivated reference set member may need to be replaced by a new member. For example, if the inactivated description was being used as a preferred term in a language reference set, then an alternative preferred term should be defined for the associated concept.
Description additions	**Extension Impact**
A new description has been added to the International Edition, and the extension specifies the acceptability of all international descriptions.	A new member of the relevant language reference set, which specifies the acceptability of the new description, may need to be created. For more information, please refer to 5.4.6.1 Add Members to a Reference Set.

Reference set inactivations	Extension Impact
The concept identifying a reference set, which is used in the extension, is inactivated.	If the reference set is not used in the extension, then no action is required. If the reference set is used in the extension, a decision should be made as to whether or not the reference set is still required. If the reference set is required, then a new local version of the reference set may need to be created. For more information, please refer to 5.4.5.1 Create New Reference Set in an Extension.
A member of a reference set, which is used in the extension, is inactivated.	If the extension uses extension reference set members to adapt an international reference set, a decision should be made as to whether or not the inactivated member is required in the extension. If it is, then the inactivated reference set member may need to be reactivated in the extension. If not, then no action is required. For more information, please refer to 6.3.2 Authoring Reference Sets.
Reference set additions	**Extension Impact**
A new reference set is added to the International Edition.	A decision should be made as to whether or not the new reference set could replace any national or locally created reference sets. If so, then this could alleviate the associated maintenance responsibilites. For more information, please refer to 6.3.2 Authoring Reference Sets.
One or more members have been added to a reference set in the International Edition.	If the extension uses extension reference members to adapt an international reference set, a decision should be made as to whether or not the new reference set members should be adopted in the extension reference set. For more information, please refer to 6.3.2 Authoring Reference Sets.

5.7.1.3 Updating Extension

Updating an extension involves creating new versions of components and reference set members that are required in the extension. Please refer to 5.4 Authoring for more information about principles and process for adding, modifying and inactivating content in an extension. In some situations, content may also need to be promoted from the extension into the International Edition (or another extension on which it depends).

Update Module Dependencies

When a new version of an extension is created, new rows must be added to the module dependency reference set to reflect the dependencies of the new version of the extension. The effectiveTime and the *sourceEffectiveTime* of the added rows must be set to the date of the new extension release.
The *targetEffectiveTime* must be set to the version of the module on which the extension modules depends (as identified by the *referencedComponentId*).

Updates to the module dependency reference set are required:

- For all existing rows in the module dependency reference set, to indicate that the dependency now applies to the new version of each module - ie the *sourceEffectiveTime* must be updated to the date of the extension release. If the extension module is now dependent on a new version of the target module, then the targetEffectiveTime must also be updated.

- For any module dependency that is no longer applicable. In this case, the relevant member of the module dependency reference set should be inactivated.
- For any new module dependency that applies to the new version of the extension. In this case, a new reference set member must be added to the module dependency reference set.

5.7.2 Change Requests

Updating the extension to accommodate user requests for additions or changes is an important aspect of the overall extension maintenance process. Consumers of extensions need to be able to submit requests for changes, and the extension producer is responsible for providing a mechanism to collect and process these change requests. This process in turn supports the continued relevance and quality of the extension itself.

National Release Centers (NRCs) are responsible for collecting requests for changes detected by Affiliates. Organizations in Member countries or territories should therefore submit their content requests to the relevant NRC. The NRC will subsequently make a decision as to the relevance of the content for the International or National Edition, as described in 5.3 Assessing Requests.

The type of service implemented to support change requests will depend on the scope and size of the extension, the anticipated number of extension consumers, and a range of other factors. In situations in which a small number of extension consumers are expected, a simple email service may be sufficient to enable consumers to request content changes. Extension producers with a greater number of anticipated consumers may decide to implement a dedicated system to support the tracking and management of change requests. SNOMED International has a formal procedure for submitting change requests, as described in the CRS User Guide.