|  |
| --- |
| IHTSDO Release configuration |
| Date:  | 20151015 |
| Version:  | 0.03 |
| Status: | Draft |

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Editor** | **Comments** |
| 0.01 | 20150507 | AAT | First draft for comments |
| 0.02 | 20150618 | AAT | Update after RTU comments |
| 0.03 | 20151015 | AAT | Updated with new proposals |

Approval History

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Approver** | **Comments** |
|  |  |  |  |

Future Review Timetable

|  |  |  |
| --- | --- | --- |
| **Review date** | **Responsible owner** | **Comments** |
| n/a | Release Manager | To be reviewed as and when enhancements are identified. |

© International Health Terminology Standards Development Organisation 2015. All rights reserved.

Table of Contents

1 Introduction 5

1.1 Background 5

1.2 Purpose 5

1.3 Scope 5

1.4 Audience 5

1.5 Major Stakeholders 5

2 Release Status configuration 6

2.1 Definition 6

2.2 Standard Configuration 8

2.3 Standard Configuration – example 10

3 Release versioning 11

3.1 Definition 11

3.2 Versioning format 11

3.3 Versioning format examples 12

3.4 Examples of versioned package names 13

4 Release Package configuration 14

4.1 Definition 14

4.2 Data files 14

4.3 Release Notes 14

4.4 Licensing statements 14

5 Release distribution 15

6 Product Templates 16

6.1 International Edition 16

6.2 Spanish Edition 16

6.3 Individual Refset Editions 16

7 Glossary of Terms 17

#  Introduction

## Background

A Service Catalog is a collection of all business and IT services that are provided by an organization. The IHTSDO Service Catalog is expanding at a significant rate, with requirements for numerous new products and services being submitted continuously.

## Purpose

The IHTSDO Release configuration document defines the standard file configurations at any given stage in the Release lifecycle.

## Scope

The Release configuration document covers the entirety of the delivery lifecycle, from the Alpha stage through to the final Production stage.

## Audience

This document is intended primarily for use by IHTSDO staff and the SNOMED CT user community.

## Major Stakeholders

In addition to the stakeholders within the user community, all IHTSDO staff have some level of involvement in this process, as at any point they may require a change to our Products and Services.

However the main stakeholders are as follows:

* Head of Technical Services
* Head of Content (Business Service Executive)
* Head of Terminology
* Release Manager

# Release Status configuration

## Definition

Release Status applies to collections of SNOMED CT release files that represent a proposed addition of components and/or derivative products to the SNOMED CT International Edition or to other items in the IHTSDO Service Catalog.

The Release Statuses are defined as follows:

|  |  |
| --- | --- |
| **Release Status** | **Definition** |
| Alpha | Previously known as the “Technology Preview” Release status, this applies to a collection of SNOMED CT release files that represent a proposed addition of components and/or derivatives to the SNOMED CT International Release or to other items in the IHTSDO Service Catalog. The Alpha status indicates the releasing party (IHTSDO or the owner of an Extension) is releasing these additional components or derivatives for review and testing by implementers and other stakeholders. 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 may be made to address the feedback received, and issues identified by testing. **Note:** The significance of an Alpha release is that this data should not be used in an operational environment that may incorporate the data into a record or create a dependency on continued maintenance of the additional components or SNOMED CT derived products.  |
| Beta | Previously known as either “Beta” or “Candidate Baseline” Release status, this applies to a collection of SNOMED CT release files that represent the final, formally endorsed release of additions of components and/or derived products to the SNOMED CT International Release or to other items in the IHTSDO Service Catalog. The Beta status indicates the releasing party (IHTSDO or the owner of an Extension) expects to subsequently confirm this Edition 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 another version of the Beta release, in order to address the issue. The releasing party need not commit to this being an actual Production release until shortly before the due date for the next release. **Note:** The significance of the Beta status is that anyone implementing this data must be prepared for withdrawal or significant changes that may occur to the additional components or derived products. Therefore, this data should not be used in an operational environment in ways that create a dependency on continued maintenance of the additional components or derivatives. However, a Beta release may be confirmed as a Production Edition and, in that case all subsequent updates to the additional components and derivatives will be fully version tracked from the release of the Beta Edition. |
| Production | This Release status applies to a collection of SNOMED CT release files that represent the final formally endorsed release of additions of components and/or derivatives to the SNOMED CT International Release or to other items in the IHTSDO Service Catalog.The Production status indicates the releasing party (IHTSDO or the owner of an Extension) commits to maintain the release history of this release and all subsequent updates. Thus from the first Production release onwards, the historical audit trail will be maintained throughout the Product’s lifetime. **Note:** The significance of the Production status is that it represents the authoritative release of the product, and implementers can use the additional components and derived products in operational systems with confidence in the subsequent maintenance of the product. |

## Standard Configuration

The standard configuration of the Release files for each Release status, are as follows:

|  |  |
| --- | --- |
| **Release Status** | **Standard Configuration** |
| **Alpha** | Module ID | Technology Preview module ID – 705115006|Technology Preview module (core metadata concept)| |
|  | Effective Date | Date on which the Alpha release is to be published. |
|  | History | No expectation of historical data present or to be traceable. |
|  | Continuity of ID’s | No requirement for UUID’s / SCTID’s to be traceable through to Production.***\* For this reason, IHTSDO strongly recommends that Alpha packages are not to be used in Production systems at any time*** |
|  | Metadata  | All metadata files for the minimum required package structure for the product must be present, but not necessarily populated (please see the “Product Templates” section below for further details of the minimum required package structure for each Product). Where metadata is present, it can be non-production data, but must have an effective time equal to the expected Production release date. |
|  | Feedback loops | As many feedback loops and subsequent updated releases of this status can be implemented, as required. |
|  | “X” Prefix for the files? | The files within the release package will be prefixed with “x”, in order to reinforce the assertion that the data should not be used in an operational environment in ways that create a dependency on continued maintenance of the additional components or derivatives. |
|  | “X” Prefix for the package name? | The release package name will be prefixed with “x” again to reinforce the prevention of use in production clinical systems. |
| **Beta** | Module ID | Agreed Production module ***Note:*** *The agreed Production Module will often be the core module (900000000000207008), but must be whatever module is agreed with the project team and documented in the Product documentation to be the module to use once in Production* |
|  | Effective Date | Date on which the Beta release is to be published. |
|  | History | Historical data should be present and traceable where possible. |
|  | Continuity of ID’s | UUID’s / SCTID’s to be traceable through to Production wherever possible. This is not, however, guaranteed to be the case.***\* For this reason, IHTSDO strongly recommends that Beta packages are not to be used in Production systems at any time*** |
|  | Metadata  | All metadata files for the minimum required package structure for the product must be present, but not necessarily populated (please see the “Product Templates” section below for further details of the minimum required package structure for each Product). Where metadata is present, it can be non-production data, but must have an effective time equal to the expected Production release date. |
|  | Feedback loops | Feedback loops and subsequent updated releases of this status should only be implemented if a critical issue is identified. |
|  | “X” Prefix for the files? | The files within the release package will be prefixed with “x”, in order to reinforce the assertion that the data should not be used in an operational environment in ways that create a dependency on continued maintenance of the additional components or derivatives. |
|  | “X” Prefix for the package name? | The release package name will be prefixed with “x”, again to reinforce the prevention of use in production clinical systems. |
| **Production** | Module ID | Agreed Production module ***Note:*** *The agreed Production Module will often be the core module (900000000000207008), but must be whatever module is agreed with the project team and documented in the Product documentation to be the module to use once in Production* |
|  | Effective Date | Date on which the Production release is to be published. |
|  | History | Historical data should be present and traceable where applicable. |
|  | Continuity of ID’s | Must be maintained for all Production releases. |
|  | Metadata  | All metadata files for the minimum required package structure for the product must be present, and should have an effective time equal to the expected Production release date. |
|  | Feedback loops | Feedback loops and subsequent updated releases of this status will be managed through the Change management process for content, and will therefore be included in subsequent Production releases. |
|  | “X” Prefix for the files? | The files within the release package will not be prefixed with “x”. |
|  | “X” Prefix for the package name? | The release package name will not be prefixed with “x”. |

## Standard Configuration – example

For a new Product, that is distributed as a single reference set, dependent on the January 2015 International Edition, but published as an Alpha release on 31st March 2015 with the intention to publish to Production on 31st August 2015, the standard configuration for the Release files for each Release status might be as follows:

|  |  |
| --- | --- |
|  | **Release Status** |
|  | **Alpha**  | **Beta**  | **Production** |
| **Module ID** | 705115006| Technology Preview module (core metadata concept)| | 900000000000207008|  SNOMED CT core module (core metadata concept)|ORA new Module defined specifically for the new derivative product | 900000000000207008|  SNOMED CT core module (core metadata concept)|ORA new Module defined specifically for the new derivative product |
| **Effective Date** | 20150331 | 20150714 | 20150831 |
| **History** | No | Yes (where possible) | Yes |
| **Continuity of ID’s** | No | Yes (where possible) | Yes |
| **Metadata**  | Files created, but not necessarily populated (see section 4.2 below for further details) | Concept, Description, Language, Stated Relationship, Inferred Relationship and ModuleDependency files created, and populated with data with an effective time of 20150731 | Concept, Description, Language, Stated Relationship, Inferred Relationship and ModuleDependency files created, and populated with data with an effective time of 20150731 |
| **Feedback loops** | 4 | 1 | N/A |
| **“X” Prefix for the files?** | Yes | Yes | No |
| **“X” Prefix for the package name?** | Yes | Yes | No |

# Release versioning

## Definition

Release versioning will reflect, within each Product, the relative Status of the release package (broken down by major and minor statuses), plus the revision number within that Status, as follows:

**[Product name]\_[Versioning format]**

 (eg) SnomedCT\_ICNPRelease\_INT\_1.1.0

## Versioning format

The format adheres to the following conventions, which are based on the best practice methodology [“Semantic Versioning 2.0.0” (“SemVer”)](http://semver.org/):

**[Major version].[Minor version].[Patch version]**

* **[Major version]** increments when incompatible changes are made to the product
* **[Minor version]** increments when backwards-compatible changes are introduced
* **[Patch version]** increments when backwards-compatible defect fixes are introduced

Additional labels for pre-release statuses are then embedded as extensions to the Major.Minor.Patch format, denoted by appending a hyphen and a series of dot separated identifiers immediately following the patch version, as follows:

**[Major version].[Minor version].[Patch version]–[pre-release status]**

In IHTSDO Release Management terms, we have adapted the Semantic Versioning 2.0.0 framework as follows:

* The “public API” as discussed in the SemVer documentation, represents for IHTSDO the combined signature of all artefacts within the Product release package. So the particular version of the RF2 format that is specific to that Product defines our “public API” – for example, the International edition contains standard RF2 files, whereas a derivative product may introduce a variation of this format. The Major version, therefore, will increment on each change of this “public API” for that particular Product.
* “Alpha” and “Beta” will follow the conventions defined above in section 2.1, and will be denoted by a “0” Major version, as they are considered still to be in the development phase of the Product, and therefore not for use in Production systems.

***Note:*** *The versioning format considers new scheduled releases of the same Edition to constitute a continuation of the Minor version numbering. This is because we consider the significant content changes in each subsequent release of a Product to constitute a new Minor version change. So where the initial Production release of the January 2016 International Edition would be allocated the version 1.0.0, the subsequent July 2016 International Edition release will not be 1.0.1, but will in fact be 1.1.0*

## Versioning format examples

|  |  |
| --- | --- |
| **Product status** | **Version** |
| A first time Production release of a new product  | 1.0.0 |
| A subsequent Production release of the same product with the same Release date, which introduces backwards-compatible changes to version 1.0.0 – this could either be due to expected content changes between, for example, January and July releases, or because of changes to the format (the “public API”) of the Product itself. | 1.1.0 |
| A subsequent Production release of the same product with the same Release date, which introduces backwards-compatible patches to resolve defects found in version 1.0.0 | 1.0.1 |
| A subsequent Production release of the same product with the same Release date, which introduces incompatible changes to the content/format (“public API”) in version 1.0.0 (for example, the removal of a derivative product, or incompatible change to the hierarchy) | 2.0.0 |
| A subsequent Production release of the same product with a new Release date | 1.0.0 |
| A subsequent Production release of a new product with the same Release date | 1.0.0 |
| A first time Alpha release of a new/existing product with new Release date | 0.1.0-alpha |
| A subsequent Alpha release of the same product on it’s twelfth patch revision (after multiple rounds of feedback identifying defects) | 0.1.12-alpha |
| A subsequent Alpha release of the same product on it’s third minor revision (after multiple rounds of feedback identifying significant changes, and potentially some prior defects which were denoted as part of the previous minor versions - 0.2.1-alpha, 0.2.2-alpha…) | 0.3.0-alpha |
| A subsequent Alpha release of the same product after multiple rounds of feedback identifying firstly significant changes required to the content, and then 6 defects to be fixed within the latest minor version | 0.3.6-alpha |
| A first time Beta release of a new/existing product with a new Release date | 0.1.0-beta |
| A subsequent Beta release of the same product on it’s fifth patch revision (after multiple rounds of feedback identifying defects) | 0.1.5-beta |
| A subsequent Beta release of the same product on it’s first minor revision (after multiple rounds of feedback identifying significant changes, and potentially some prior defects which were denoted as part of the previous minor versions - 0.1.1-beta, 0.1.2-beta…) | 0.2.0-beta |
| A subsequent Beta release of the same product after multiple rounds of feedback identifying firstly a significant change required, and then eight defects to be fixed within the that first minor version | 0.2.8-beta |

## Examples of versioned package names

|  |  |
| --- | --- |
| **Product status** | **Release Version** |
| A first time Production release of a new International edition product | SnomedCT\_RF2Release\_INT\_20160131\_1.0.0 |
| A first time Production release of an existing International edition product | SnomedCT\_RF2Release\_INT\_20160131\_1.0.0 |
| A subsequent Production release of the same International product with the same Release date, due to a required content/format change | SnomedCT\_ RF2Release \_INT\_20160131\_1.1.0 |
| A subsequent Production release of the same International product with the same Release date, due to a required defect patch | SnomedCT\_ RF2Release \_INT\_20160131\_1.0.1 |
| A first time Alpha release of an existing International edition product | SnomedCT\_RF2Release\_INT\_20160131\_0.1.0-alpha |
| A first time Beta release of an existing International edition product | SnomedCT\_RF2Release\_INT\_20160131\_0.1.0-beta |
| A first time Production release of a new derivative product  | SnomedCT\_GPFP\_INT\_20180731\_1.0.0 |
| A subsequent Production release of the same derivative product with the same Release date, due to a required content/format change | SnomedCT\_GPFP\_INT\_20180731\_1.1.0 |
| A subsequent Production release of the same derivative product with the same Release date, due to a required content/format change, and three subsequent required defect patches | SnomedCT\_GPFP\_INT\_20180731\_1.1.3 |
| A subsequent Production release of the same derivative product with a new Release date | SnomedCT\_GPFP\_INT\_20190131\_1.0.0 |
| A first time Alpha release of a new/existing derivative product with new Release date | SnomedCT\_LOINC\_INT\_20160801\_0.1.0-alpha |
| A subsequent Beta release of the same derivative product on it’s third revision | SnomedCT\_LOINC\_INT\_20160801\_0.1.3-beta |
| A subsequent Beta release of the same derivative product with a new Release date | SnomedCT\_LOINC\_INT\_20170501\_0.1.0-beta |

# Release Package configuration

## Definition

This section describes the configuration of the distribution package that contains all of the Release artifacts, including data files, release notes, etc.

##  Data files

The full complement of core data files will be included in all International and derivative Editions, even where there is no data in the file. So, for example, in a release of a new derivative map product, the expected files to be populated are only the SimpleMap, ModuleDependency and RefsetDescriptor files – however, all remaining standard files are included, with only headers present.

This is to demonstrate that these files have not been omitted in error, but have been considered and are intentionally empty.

Please see the section below on “Product Templates” for details of the files that are included in each product.

## Release Notes

The Release Notes have historically been included in the Release package itself, as .pdf documents.

They will now be setup as a web page on Confluence, with a link to this page included both in the Release announcement, plus the Readme file inside the Release package (for those accessing the Release package via MLDS). The Release Notes will be hosted on an open page, allowing all members and affiliates to access them, but also anyone who might be interested in becoming a member in future.

The benefits of this are to allow everyone access to the Release Notes without having to download the entire Release package, and to provide a clear audit trail of all Release Notes, enabling easy analysis of the historical changes introduced.

## Licensing statements

Licensing statements have historically been included in the Release Notes, and in the Readme files.

They will now be setup on separate Confluence pages (one per licensing agreement), which will then be linked to via the Release Notes and Readme files.

This enables us to maintain the license statements without re-publishing entire Releases, thus ensuring that the statements remain up to date at all times.

# Release distribution

The Release packages will be distributed via two separate systems, Confluence and MLDS:

|  |  |  |
| --- | --- | --- |
|  | **Confluence** | **MLDS** |
|  |  |  |
| **Audience** | IHTSDO Members | IHTSDO Affiliates |
| **Access rights** | Members and affiliates will be allocated the relevant access rights.There are 3 levels of permissions:1. *Unrestricted* other than requiring IHTSDO Confluence login (for Tech Previews without License restrictions only)
2. *Restricted* by person (for all Releases – the people who will be allowed to view/download decided on a product-by-product basis)
3. *Private* (internal IHTSDO use only)

Some pages (for example those hosting the Release Notes), will also be left without any restrictions, in order to allow everyone access to view the relevant information. | Non-member affiliates need to apply for a user account, at which point they will have access to all Release packages hosted on MLDS. |
| **Content** | The content hosted on Confluence will include, but is not restricted to:* IHTSDO Service catalog
* Release Schedule
* Release packages
* Release notes
* Issue trackers
* Known Issues
* Other documentation
 | From a distribution perspective, MLDS will host Release packages only. |
| **Communication** | The Release announcements will be published on a blog-style page on Confluence. Members can then apply for access to this page, which will allow them to view the announcements and receive alerts when new announcements are posted. | Users of MLDS will receive automated updates when each new Release package is posted. |

# Product Templates

The Product Templates detail the minimum expected set of files for each Product, plus the folder structure in which they should be packaged.

## International Edition

Please download the template from the following link:

[https://confluence.ihtsdotools.org/download/attachments/6161107/manifest - TEMPLATE INTERNATIONAL.xml?version=1&modificationDate=1431015236934&api=v2](https://confluence.ihtsdotools.org/download/attachments/6161107/manifest%20-%20TEMPLATE%20INTERNATIONAL.xml?version=1&modificationDate=1431015236934&api=v2)

## Spanish Edition

Please download the template from the following link:

[https://confluence.ihtsdotools.org/download/attachments/6161107/manifest - TEMPLATE SPANISH.xml?version=1&modificationDate=1431015269263&api=v2](https://confluence.ihtsdotools.org/download/attachments/6161107/manifest%20-%20TEMPLATE%20SPANISH.xml?version=1&modificationDate=1431015269263&api=v2)

## Individual Refset Editions

Please download the template from the following link:

[https://confluence.ihtsdotools.org/download/attachments/6161107/manifest - TEMPLATE INDIVIDUAL REFSETS.xml?version=1&modificationDate=1431015282625&api=v2](https://confluence.ihtsdotools.org/download/attachments/6161107/manifest%20-%20TEMPLATE%20INDIVIDUAL%20REFSETS.xml?version=1&modificationDate=1431015282625&api=v2)

# Glossary of Terms

|  |  |
| --- | --- |
| **Term** | **Meaning** |
| Change | The addition, modification or removal of anything that could have an effect on IHTSDO Services (excluding Service Requests) |
| Configuration Item (CI) | A component of the infrastructure (entire system or single component) or an item associated with the infrastructure, such as documentation, a piece of software, or an individual data item, that is under the control of configuration management.  |
| Customer(s) | Consumers of IHTSDO services, both internal and external. |
| Definitive Software Library (DSL) | The library in which the definitive versions of all software products are stored & protected.  |
| Forward Schedule of Change (FSC) | Schedule that contains all changes approved for implementation and their proposed implementation dates. |
| Incident  | Any event that is not part of the standard operation of a service that causes, or may cause, an interruption to, or a reduction in, the quality of service. |
| Known Error (KE) | A problem for which the root cause is known and a temporary workaround or a permanent alternative has been identified. Known errors remain unless they are permanently fixed by a Change. |
| Key Performance Indicators (KPI’s) | These are the criteria that are used to measure and report on the quality of the deliverables both during and after the project. |
| Management information (MI) | This can take many forms, most usually progress reports and performance against quality metrics, in order to provide the Management Team with visibility of the current status of the project, allowing them to take corrective action wherever necessary. |
| Permanent Package | This is a Permanent Edition of a SNOMED CT content package, which will remain static throughout subsequent releases, and will therefore only be published once. It will, however, remain permanently available for download by all approved customers. |
| Procedure | A document containing steps that specify how to achieve an activity. Procedures are defined as part of Processes. |
| Process | A structured set of Activities designed to accomplish a specific Objective. A Process takes one or more defined inputs and turns them into defined outputs. A Process may include any of the Roles, responsibilities, tools and management Controls required to reliably deliver the outputs. A Process may define Policies, Standards, Guidelines, Activities, and Work Instructions if they are needed. |
| Process Owner | A Role responsible for ensuring that a Process is Fit for Purpose. The Process Owner’s responsibilities include sponsorship, Design, Change and continual improvement of the Process and its Metrics.  |
| Product Owner | The Product Owner is responsible for ensuring that the Product is Fit for Purpose and Fit for Use by the community. They will have active involvement in, and take ultimate responsibility for the design, build, testing and deployment of the Product (though other roles will usually action these tasks on the Product Owners behalf). |
| RACI Matrix | This is a cross reference table that identifies within the various stakeholders involved in the project, who will be Responsible, Accountable, Consulted and Informed regarding the various stages of the project. |
| Release | A collection of new or changed Configuration Items (CIs), which have been tested and introduced together into the production environment. |
| Release Validation Framework (RVF) | An IHTSDO application that validates all RF1 and RF2 SNOMED CT files to the specified level, and outputs warnings of all non-compliance. |
| Request for Change (RFC) | A form, or screen, used to record details of a request for a change to any Configuration Item (CI) within the infrastructure that is under the control of configuration management. |
| Requestor | This is the person or persons who Request the Change to the service (whether this be an update to, or a new Product). This is almost always the same person as the Product Owner, but in some cases someone may act as a proxy for the Product Owner when requesting the Change. |
| Service Acceptance (SA) | A formal process that leads to the acceptance of a new or significantly changed IT Service. |
| Service Acceptance Criteria (SAC) | A set of criteria used to ensure that an IT Service meets its functionality and quality requirements and that the IT Service provider is ready to operate the new IT Service when it has been implemented.  |
| Service Catalogue | Documented details of all IHTSDO services and products provided, including key features. This catalogue will form the basis of understanding of all services offered, their components, features and charges, etc.  |
| Service Level Agreement (SLA) | Service Level Agreement is a written agreement between the service provider and the customer that documents agreed service levels for a service |
| Snomed Release Service (SRS) | An IHTSDO application that processes Snomed CT (and other) input files, and exports a complete Release package. |
| Stakeholders | People who have an interest in the product being implemented. Stakeholders may be interested in the activities, targets, resources or deliverablesof a product**.** |
| Use Case | A use case is a list of steps, typically defining interactions between a role and a system, in order to achieve a goal. The role can be a human, an external system, or time. The system is something being developed (software, data products or documents, etc), with which the role interacts. The use case then defines a sequence of actions that yields an observable result.  |