# 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<br>• Description file conforms to the structure specified in the Description File Specification<br>• Relationship file conforms to the structure specified in the Relationship File Specification<br>• 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<br>• New delta file consists of all rows from the new full release which has an effectiveTime equal to the new release date<br>• 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<br>• 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.<br>• Primary keys are unique within each file<br>○ For files in the full release, the combination of the *id* and the *effectiveTime* is unique<br>○ 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:<br>○ An active description of type \| Fully specified name\|<br>○ At least one active description of type \| Synonym\|<br>○ At least one active \| is a\| relationship<br>○ A transitive \| Is a\| relationship to the root concept<br>○ A *definitionStatusId* which refers to an active descendant of \| Definition status\|<br>• All active relationships must conform to the following rules:<br>○ *sourceId* and *destinationId* must both refer to active concepts<br>○ *typeId* must refer to an active descendant of \| Attribute\|<br>○ *characteristicTypeId* and *modifierId* must refer appropriate active metadata concepts<br>• All active descriptions must conform to the following rules:<br>○ *conceptId* must refer to a valid concept (which may be active or inactive)<br>○ *typeId* and *caseSignificanceId* must refer to appropriate active metadata concepts |