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