

UTG Versioning Refinement (Not Yet Approved)

This page outlines the approach for versioning in UTG. A versioning approach is needed to unify how versions are assigned and maintained going forward in UTG. Currently V2, V3 content etc have used different approaches and syntax. The versioning should provide information to inform users to the severity of the change(s) and if they need to update their content.

NOTE: This page will be used for discussing and voting on versioning requirements with the Vocabulary WG. Approved requirements are noted on [Documented Execution of UTG Versioning](#).

The UTG Versioning option detailed below was proposed and approved by the Vocabulary WG ([2019-10-21 Vocab WG Call Agenda/Minutes](#)).

SCOPE: "Versioning" here refers to the versioning of UTG Resources (Code System, Value Set, Naming System and Concept Map). Thus, any reference to backwards compatibility is in reference to the content, and is not in reference to any API.

To see a summary of all options considered for versioning, please go to [UTG Versioning Planning and Discussions](#).

UTG Versioning

1. Advantages
 - a. Easier to automatically apply than Semantic Versioning
2. Disadvantages
 - a. Almost everything is a major change. How does this align with the use of GitHub branches for UTG now?
 - b. Date Time Stamp format DOES NOT align with this approach, but is not precluded from being part of the metadata associated with each version
3. Possible specification: The following information is only the beginning of a much longer discussion that needs to occur in order to provide detailed constraints about what constitutes a major, minor, or technical change
 - a. For **CodeSystem**:
 - Major = Adding, Deleting, and any concept Status changes (to/from Active to Retired, ActiveBackwards Compatible, or DeprecatedBackward Compatible) AND any defining Concept changes (hierarchical changes, defining relationships), changes to properties that are defining (to be listed - such as isSelectable), or changes to <name>
 - Minor = Concept attribute changes (non-defining changes/properties - such as synonymCode), concept Status change of Active/Deprecated, addition of language translations of the text bits (print names, descriptions, etc.), or changes to CodeSystem metadata such as <title>, <Committee>, <Steward>, or <Publisher>
 - Technical Correction = Encoding changes, typos
 - Noting anything beyond Major would be a different CodeSystem (such as changes to OID and Defining URI)
 - Everything here applies to Supplements and Fragments as well
 - Open question: What about concept name and term name changes? – Reuben believes these would constitute a major change
 - b. For **ValueSet**:
 - NOTE: Expansions are not considered here, only Value Set Definitions.
 - Per VSD, for Value Set Definition, any change to definitional elements means a new Value Set Definition (not version). This includes scope and immutable and any changes to Defining URI or OID. Any change to non-definitional elements do not require a new Value Set Definition.
 - Per VSD, for Value Set Definition Version, any change to definitional elements means a new Value Set Definition Version must be created with a new Value Set Version Identifier assigned but the Value Set Identifier does not change.
 - NOTE: It appears that the VSD is silent on the impact of non-definitional element changes on the Value Set Definition Version identifier (section 5.2.3.1). It was also identified that a typo exists in the aforementioned section usage notes (Value Set Definition should be replaced with Value Set Definition Version).
 - i. Definitional elements include:
 1. Content Logical Definition (FHIR valueset.compose)
 - ii. Non-Definitional elements that would not trigger a Version change (per VSD)
 1. Activity Status
 2. Activity Status Date
 3. Workflow Status
 4. Steward, Author, Comments, CommentString, CommentTimeStamp, TrustedValueSetExpansionSource, Type
 - Major = Any change to the Content Logical Definition (CLD) - Note, underlying concept status changes in the code system do NOT inherently change the CLD, only the expansion
 - Minor = Any non-definitional element changes to either the Value Set Definition or Value Set Definition Version, typo in definitional elements
 - Technical Correction = Encoding changes, typos in non-definitional elements
 - Need to distinguish when change to CLD should result in new Value Set, need to consider changes in scope
 - Scope change = new Value Set

Code Systems

1. The version for the code system objects will be in <version>
2. Note this is for internal code systems, ie those owned, authored, maintained and published by HL7 only. It does include the HL7 CS stubs for external 'hooks'.
3. Based on discussions in conference calls and with Lloyd on impact of versioning number assignment for initial versions, we will do the following for Code Systems
 - a. For all v3 code systems, the initial version will be 2.0.0
 - i. At some point in the future, if needed we will figure out how to mine the version information out of the last modified change tracker in the v3 maintenance mdb file, but will NOT be done unless needed
 - b. For all v2 code systems, the initial version will be 2.n.0, where 'n' is the integer of the code system version from the v2 mdb file.
 - c. For all FHIR code systems, the initial version will be 4.2.0. (may be done already as of 1/6) Note this is the CI version in FHIR as of 2/21/2020; we need to discuss in Sydney what we should really use as a starting version for the UTG FHIR code systems.

- i. Not yet decided what will happen when FHIR R5 is released
 - ii. Not sure yet if we need to do something different for the normative code systems in R3 and R4
 - iii. Brings up the older versions question for the FHIR code systems
4. For Code System updates:
- Major = Adding, Deleting, and any concept Status changes (to/from Active to Retired, ActiveBackwards Compatible, or DeprecatedBackward Compatible) AND any defining Concept changes (hierarchical changes, defining relationships), changes to properties that are defining (to be listed - such as isSelectable), or changes to <name>
 - Minor = Concept attribute changes (non-defining changes/properties - such as synonymCode), concept Status change of ActiveDeprecate d, addition of language translations of the text bits (print names, descriptions, etc.), or changes to CodeSystem metadata such as <title>, <Committee>, <Steward>, or <Publisher>
 - Technical Correction = Encoding changes, typos
- Noting anything beyond Major would be a different CodeSystem (such as changes to OID and Defining URI)
 Everything here applies to Supplements and Fragments as well
Open question: What about concept name and term name changes? – Reuben believes these would constitute a major change

Summary of Execution Details

The table below will be populated and discussed by the UTG and Vocabulary WGs. Items in blue text indicate that the element is not currently used in UTG.

Technical Correction will be used for encoding changes or fixing typos.

Element Name	Change Type
CodeSystem.Url	New Code System (for HL7 created CS; externals have different rules; see HTA/Vocab guidance on changing URLs for external code systems)
CodeSystem.Identifier	New Code System (if an OID or V2 mnemonic; externals have different rules see above)
CodeSystem.Version	N/A
CodeSystem.Name	Major
CodeSystem.Title	Minor
CodeSystem.Status	Major
CodeSystem.Experimental	Not Currently In Use for UTG
CodeSystem.Date	Not Currently In Use for UTG
CodeSystem.Publisher	Minor
CodeSystem.Contact	Minor
CodeSystem.Description	Major unless only change is improved clarity or perform editorial corrections.
CodeSystem.UseContext	Not Currently In Use for UTG
CodeSystem.Jurisdiction	Not Currently In Use for UTG
CodeSystem.Purpose	Not Currently In Use for UTG
CodeSystem.Copyright	Minor, unless change from unencumbered to encumbered.
CodeSystem.CaseSensitive	Not Widely Used in UTG (only two objects)
CodeSystem.ValueSet	Pro-forma in UTG, Cannot Change Value
CodeSystem.HierarchyMeaning	Major
CodeSystem.Compositional	Not Currently In Use for UTG
CodeSystem.VersionNeeded	Not Currently In Use for UTG - but under consideration for ballot-bound code systems
CodeSystem.Content	N/A (header element)
CodeSystem.Supplements	Not Currently In Use for UTG
CodeSystem.Count	Not Currently In Use for UTG
CodeSystem.Filter (and sub-elements)	Not Currently In Use for UTG

CodeSystem.Property (including CodeSystem.Concept. Property.Code and .Value)	Major for Concept Status and defining property changes (like isSelectable), status changes to /from Active to Retired, ActiveBackwards Compatible, addition of language translations of the text bits (print names, descriptions, etc.). Minor for non-defining property changes (like editorial property value changes). Not sure about ActiveDeprecated? or DeprecatedBackward Compatible)? (probably minor)
CodeSystem.Concept (and all sub-elements)	All changes to concepts by default are an increment of the major number. Any proposed change could include a proposal to change the minor number instead if the change is asserted to be minor in the change proposal.
concept.code	Code never changes so N/A
concept.display	Major. Typos are minor.
concept.definition	Major. Typos are minor.
concept.designation.	Major
concept.property	Major. Minor for non-defining property changes (like editorial property value changes).
Are there special or different rules about certain 'special' code systems, such as UTG concept properties, or concept domains, or V2 Tables?	Yes, handled on a case by case basis.

Value Sets

1. For value sets, we must evaluate all of the attributes defined in VSD to determine which ones will trigger what kind of version change, ie new object, increment of major number, increment of minor number.
2. VSD attribute list:
 - a. All Revision History attributes - updates to these are only from updates to other elements, so do not trigger and version ID things by themselves
 - b. Creation Information - only set once when created with version 1.0.0
 - c. Value Set Definition Naming
 - i. Value Set Definition Naming.Name - update triggers major number increment; new entry for an additional name such as in a different language, triggers minor number increment
 - ii. Value Set Definition Naming.Name Language - update triggers minor number increment
 - iii. Value Set Definition Naming.Preferred Name Indicator - update triggers minor number increment
 - d. Value Set Definition
 - i. Value Set Identifier - change to the identifier triggers new value set; addition of one or more secondary identifiers with no change to primary triggers minor number increment
 - ii. Scope - change to the scope triggers new value set, unless just editorial text change to scope text, in which case it triggers minor number increment
 - iii. Immutable - change to the flag triggers new value set
 - iv. Definition URL - change triggers minor number increment (verify with team)
 - v. Intellectual Property Information - change triggers major number increment unless change is spelling or other editorial corrections in which case it is minor number increment
 - vi. Experimental - change to flag triggers major number increment
 - vii. Source Reference - change triggers major number increment unless editorial in nature which triggers minor number increment
 - viii. Keyword - change triggers minor number increment (verify with team)
 - ix. Use - change triggers major number increment (verify with team)
 - e. Value Set Version
 - i. Comments - change or additional comments trigger minor number increment
 - ii. Version Identifier - change triggers new value set
 - iii. Activity Status - change triggers major number increment
 - iv. Activity Status Date - change triggers major number increment
 - v. Author - change triggers minor number increment
 - vi. Steward - change triggers major number increment
 - vii. Trusted Value Set Expansion Source - change triggers major number increment
 - viii. Type - change triggers major number increment
 - ix. Workflow Status - change triggers major number increment
 - f. Content Logical Definition
 - i. Locked Date - change triggers major number increment
 - ii. ActiveOnly - change triggers major number increment
 - iii. CLD Syntax Reference - change triggers major number increment
 - iv. Content Expression - if change supports scope change and the coverage or granularity of the value set is changed, then it is a new value set. If the syntax is modified to improve efficiency of processing or simply the syntax but the intent is unchanged in terms of coverage and granularity then it is a major number increment. Changes to code system basis is a new value set (verify with team).
 - v. Non-computable syntax expression - change is either a new value set or a major change depending upon the interpretation (verify with team)

Summary of Execution Details

Element Name	Change Type
--------------	-------------

RevisionHistory	Does not trigger change
ValueSetDefinitionNaming.Name	Minor
ValueSetDefinitionNaming.NameLanguage	Minor
ValueSetDefinitionNaming.PreferredNameIndicator	Minor
ValueSetDefinition.ValueSetIdentifier	Change = New value set Addition of secondary = Minor
ValueSetDefinition.Scope	Change = New value set Editorial change = Minor
ValueSetDefinition.Immutable	Change = New value set
ValueSetDefinition.DefinitionUrl	Minor (check with team)
ValueSetDefinition.IntellectualPropertyInformation	Major Editorial change = Minor
ValueSetDefinition.Experimental	Major Editorial change = Minor
ValueSetDefinition.SourceReference	Major Editorial change = Minor
ValueSetDefinition.Keyword	Minor (check with team)
ValueSetDefinition.Use	Major (check with team)
ValueSetDefinitionVersion.Comments	Minor
ValueSetDefinitionVersion.VersionIdentifier	Change = New value set
ValueSetDefinitionVersion.Activity Status	Major
ValueSetDefinitionVersion.ActivityStatusDate	Major
ValueSetDefinitionVersion.Author	Minor
ValueSetDefinitionVersion.Steward	Major
ValueSetDefinitionVersion.TrustedValueSetExpansionSource	Major
ValueSetDefinitionVersion.Type	Major
ValueSetDefinitionVersion.WorkflowStatus	Major
ContentLogicalDefinition.LockedDate	Major
ContentLogicalDefinition.ActiveOnly	Major
ContentLogicalDefinition.CLDSyntaxReference	Major
ContentLogicalDefinition.Content Expression	Scope change = New value set Change to code system basis = New value set (check with team) Modified to improve efficiency of processing but intent unchanged = Major
ContentLogicalDefinition.NonComputableSyntaxExpression	Change = New value set or Major (depending upon interpretation)