Vocabulary Maintenance at HL7

Submit changes to this page

Vocabulary Maintenance at HL7

This is the place to find information on the maintenance of vocabulary at HL7 including the Unified Terminology Governance (UTG) process and operations:

- sign up to watch proposals for changes in vocabulary;
- comment on proposed changes; and
- vote on proposed changed

All of the Code Systems and Value Sets for the Code Systems that are not ballot-bound are maintained through this UTG process as of Q2/Q3 2020. For a list of ballot-bound Code Systems and Value Sets, click here.

Overview

HL7 maintains multiple terminologies - v2, v3, CDA value sets and FHIR(R).

The ongoing maintenance of these terminologies is resource intensive and quite opaque to much of the community and as a result, a need for a uniform vocabulary governance process or strategy across all the HL7 product lines was identified. The growing popularity of HL7 FHIR(R) and the associated implementation guides and the increase in participation of the FHIR Community escalated the need for such a process.

There is a need to maintain the terminologies that support all of HL7’s products (v2, v3, FHIR, CDA, etc.) in a way that is responsive and improves quality while reducing the resources that both HL7 and its volunteers put into the process. As well, there’s a need for the process to align with the community’s expectations for a more modern, continuous peer-feedback related process.

The UTG project replaced previous vocabulary maintenance processes (including harmonization) for HL7 internationally published normative standards. Additional information about project details, requirements, project management, and documentation can be accessed at the Unified Terminology Governance Project (UTG) Page.

Education Materials

- Webinars
  - The HL7 Terminology - brief introduction to the HL7 Terminology (THO)
  - Terminology Governance and Publishing at HL7 - a more in depth session on the HL7 Terminology, including the representation of HL7 product family content and an introduction to the UTG process. No need to check out the first video if you watch this one, as it builds upon the first video!
  - Reviewing and Voting on HL7 Vocabulary Change Proposals - walks through UTG review and governance process for vocabulary maintenance at HL7 for the HL7 Terminology Review audience. The Terminology Governance and Publishing at HL7 Webinar is a pre-requisite.
  - Overall Process for Submitting HL7 Vocabulary Change Proposals - provides overview of change proposal submission, including how to create a change proposal ticket. The Terminology Governance and Publishing at HL7 Webinar is a pre-requisite. Future tutorials are planned to discuss the technical details of creating the content changes and using the UTG custom tooling.

  - General Architecture of UTG

Maintenance of HL7 Terminology

This section provides a high-level overview of the workflow process for submitting a proposal and participating in consensus review.

- SUBMISSION: submit a proposal for changes that contain all technical details to the vocabulary
- VALIDATION: to ensure the proposal includes required information
- CONSENSUS REVIEW: voted on by the community
- PUBLISH: approved changes by a majority vote are implemented to the the current build of the terminology
The UTG process decreases the burden on HL7 resources by having the proposal submitter create and submit the content that they needed added, modified, or deleted in their own development branch (via BitBucket directly through Jira). Proposals are submitted using a Jira-based workflow that is open to the community. Automated checks and validations ensure that the content meets proposal requirements. The proposal goes through a consensus workflow to come to a resolution. If approved, the use of the Bitbucket branching allows simplified integration into the master repository.

The sections below provide a high-level overview of the Change Proposal Submission and Consensus Review Process.

**Consensus Review**

All change proposals are validated and formatted for a consensus review. Votes are weighted and there are several requirements (i.e. votes from specific users and groups) that must be met for the content to be approved or rejected.

HL7 members wishing to participate in the Consensus Review of proposed change proposals are welcome to participate. No tooling is required to participate.

The basic steps to become a proposal reviewer and participate in Consensus Review:

1. Get access to HL7 Jira by requesting an account on [https://ji ra.hl7.org/secure/Dashboard.jspa](https://jira.hl7.org/secure/Dashboard.jspa) (if you do not already have one)
2. Request reviewer/voter permissions by filling out the following form (DO NOT OPEN IN NEW TAB):
   - Request Reviewer Permissions
3. Proposals in Consensus Review are displayed in the UTG Consensus Review Dashboard
4. Review details of proposal and view rendered version of proposed content via link on ticket
5. Use comments to discuss the proposal if necessary.
6. Vote Affirmative or Negative

Detailed step-by-step instructions can be found at: Consensus Review and Voting Process
Change Proposal Submission

The steps below describe the process for submitting a proposal. More detailed information on the process is documented at UTG Tooling and Proposal Documentation.

Don't know if you have permissions to be a Submitter of Vocabulary Change Proposals? When you create a proposal and it goes into the Environment Setup state, if you see a button on the top “Draft A Proposal” then you ARE a Submitter! If you don't see that button, you can become a Submitter by clicking this link (do not open in new tab):

Request to become UTG Submitter

The basic steps to submit a change proposal are:

1. Get access to HL7 Jira by requesting an account on https://jira.hl7.org/secure/Dashboard.jspa
2. Request Submitter permissions by filling out the following form:
   
   Request to become UTG Submitter

   a. If you are NOT an HL7 member, please add a comment in the ‘Note’ field explaining why you want to become a submitter and describe your relationship to HL7 and/or UTG
3. Install required tooling (Sourcetree)
4. Draft the proposal (written information) on the UTG Jira Project
5. Create and download a branch to edit content using Sourcetree
6. Use XML Editor to create content change(s) for the proposal - changes to the UTG resources
7. Commit new or changes resources to branch using Sourcetree and Push the changes to BitBucket
8. Submit proposal in Jira

Detailed step-by-step instructions can be found at: How To Submit a UTG Change Proposal

Setting Up Notifications for UTG Proposals

The UTG issue tracking system (JIRA) is set up so that reviewers must opt-in to subscribe to notifications.

The system has been initially configured for FIVE (5) different types of notifications or “filters”:

- Consensus Review: https://jira.hl7.org/issues/?filter=13111
- Meeting Needed: https://jira.hl7.org/issues/?filter=13114
- Rejected: https://jira.hl7.org/issues/?filter=13113
- Sent for Implementation: https://jira.hl7.org/issues/?filter=13115
- Withdrawn: https://jira.hl7.org/issues/?filter=13112

Opting-in will trigger emails to be sent to you when any change proposal enters the subscribed “State” in the UTG workflow (the names above are the State names).

Here is a helpful Jira quickstart guide on using filters/subscriptions which can help you to better understand how to use this system in a way that works best for you personally:

https://confluence.atlassian.com/jiracoreserver085/working-with-search-results-981157534.html
Create New Notification

1. In order to opt-in for one or more of these notifications, you must login to your account on [https://jira.hl7.org](https://jira.hl7.org) and go to the identified link in the list above. The link will show subscribe (or unsubscribe if you are already setup).
2. Click the filter link for the type of notification you are interested in and then click on ‘Details’ on the top-middle of the page and select ‘New subscription’.
3. Customize the notifications per your liking and click ‘Subscribe’.
4. Once you have subscribed to these notification filters, you will begin to get email messages from the HL7 webmaster account when a proposal enters the State that you subscribed too.

Opt-out of Notification

1. To opt-out, you can select that option on any of the filter links in the list above, click ‘Details’ and then ‘Manage Subscription’.
2. Click ‘Delete’ on the row of the filter(s) you would like to opt-out of.

Reporting a Vocabulary Issue

To report vocabulary issues found on the HL7 terminology pages, please follow the steps below. Note that reporting a vocabulary issue will add it to a queue of other reported issues in which content issues have been identified but no solution has been provided. Items in this queue will not be addressed as quickly as Change Proposals.

1. Get access to HL7 Jira by requesting an account on [https://jira.hl7.org/secure/Dashboard.jspa](https://jira.hl7.org/secure/Dashboard.jspa)
2. Draft the summary of the issue via a ticket on the UTG Jira Project.
   a. From the link above, click ‘Create’ in the top toolbar
   b. Ensure Project field is ‘UTG Change Proposals (UP)’
   c. For Issue Type, select ‘Change Request’
   d. Enter the Summary of the issue. This will be the title of the ticket, so make it concise.
   e. For Sponsor, enter the WG that you think is best suited to review this request. Note that this field will be removed as a requirement at this stage in the near future
   f. For Proposal Type, select the type(s) of proposals that best describe the content where the issue was identified
   g. For Proposal Description, enter a description of the content issue. Be as descriptive as possible.
   h. For Change Objects, link to the artifact(s) with the content issues being described in the current build of THO
   i. For Assignee, select ‘Unassigned’
   j. Double-check the information and click ‘Create’ at the bottom of the form.
3. Navigate to the newly created ticket
4. To submit the ticket as a content issue, click the 'Identified Content Issue' button at the top of the ticket. The Status will change to 'Waiting for Input'