**THIS DOCUMENT HAS BEEN MIGRATED TO CONFLUENCE AT THIS LOCATION:**[**https://confluence.hl7.org/display/SD/SDWG+Extension+Release+Process**](https://confluence.hl7.org/display/SD/SDWG+Extension+Release+Process)

SDWG GForge Release Process

Several years ago when we dealt with the security flaw in our stylesheet, the final step in releasing that content included creating a release package and adding it to the “Release” section of the SDWG GForge project. We’ve never done a Gforge release since then. We have been using the SVN component of GForge to keep some of our content under source control, but as has been pointed out, it’s very difficult for those without GForge accounts to get to the material (not impossible just difficult). Our SVN is set up to allow anonymous checkouts, but very few people know how to do SVN checkouts.  For those without a GForge account, retrieving a release package is much simpler via a conventional http: file download (i.e. clicking on a web link to a file for download).

Process Proposal:

* Provide SDWG GForge commit access to those who are making edits to content stored in SDWG GForge SVN
* Anyone with commit access can pull together and propose a new release of a package of content
* The content may be a single artifact such as a stylesheet or a more complex package containing multiple file such as the entire CDA schema package).
* The release package should contain a release note indicating the changes since the prior release
* The release package should be put into a zip file.
* Initially, the release package is uploaded o the SDWG project Release section on GForge via the manage packages option found here: <http://gforge.hl7.org/gf/project/strucdoc/frs/?action=index>
* If it’s the first release of the package it will have to be added as a new package (as opposed to a new release of an existing package).
* Release package numbering – Keep it simple – the first release is 1.0 with new major releases incrementing that by 1. Minor releases would be a “dot” release (1.1). If you’re not sure it’s a major or minor release, ask the co-chairs. Use the release number as the release name in GForge (a required field for a release).
* In GForge – Manage Packages, select the package you are releasing and press “Add Release” button.
* This brings up the Add Release form, which needs to be filled out. Use the release number as the release name (required). Add Release notes and document the changes. Use the current date as the release date. Towards the bottom of the file is a place to upload the release package zip file. For the initial upload, use the default status for the release as “0-Proposed”.
* Send a note to the SDWG list (copy the co-chairs) that a new release package is available for review on GForge. Provide a link and a description of the changes. You can request that reviewers let you know if they intend to review the package and how long they need to do that. Don’t be surprised if you don’t hear from anyone. In the absence of any reviewers volunteering, give it a week and then request the co-chairs add approving the release packager on a SDWG call.
* Once the review period is over, (1 week, or whatever the reviewers indicated the needed), request the co-chairs add approving the release packager as a “Alpha” release on a SDWG call.
* Once the package is approved on a SDWG call, go back to GForge and change the release development status of the Release package to “3-Alpha”.
* If at some point the release package is actually piloted and no defects are identified, then you can propose to SDWG on a conference call that the release package be promoted to “Beta” status. If that’s approved, update the status of the package to “4-Beta”. I don’t know that we want to go beyond identifying stuff as Beta. If there are any changes as a result of the pilot, then you actually need a new release. Under that circumstance, you can probably propose it being released as “Beta” immediately (instead of “Alpha”) if the changes were actually tested during the pilot.

This proposed approach to handling releases of these artifacts in GForge is a bit looser than the process we have around publication of standards. It’s designed to allow the people who are actually doing the work to drive moving this stuff forward instead of putting the co-chairs in the middle where we become bottle necks (frustrating the people who are doing the work). It does give the work group a say in the release process since work group approval is needed to move the status of a release beyond “Proposed”. It does depend on an honor system, since there is nothing preventing someone from making the status is Beta, etc.