

Semantic Versioning

*A proposal for a clearer release artifact
naming strategy*



 @djvreeman

Daniel J. Vreeman, PT, DPT, MS

Regenstrief-McDonald Scholar in Data Standards
Indiana University School of Medicine

Director, LOINC and Health Data Standards
Regenstrief Center for Biomedical Informatics



INDIANA UNIVERSITY

SCHOOL OF MEDICINE

Department of Medicine



Regenstrief Institute

Better Care. Better Health.

Background

We've been using numeric version numbers for LOINC (e.g. 2.61) and RELMA (6.12) in the major.minor style.

We haven't really said anything about our version naming approach.

Addressed some implementation issues by designing the LOINC Table Core file.

We've had examples of both LOINC and RELMA where we had to publish "patch" releases.

Some problems

Haven't defined what constitutes major/minor (or patch) changes

Content vs. file structure changes

Now have lots of accessory files and supplemental artifacts that have their own developmental path, but depend on an underlying LOINC version.

Proposed Solution:

Semantic Versioning

Informed by [FHIR's versioning approach](#)

Define major, minor, and patch updates in our context according to the impact on users (“breaking” vs “non-breaking”).

For accessory files, separate out the “version of LOINC” this depends on from version of the artifact

Use these in version labels in artifact names.

Overall Naming Pattern

```
LOINC_<LoincVersion>_<ArtifactLabel>_<ArtifactVersion>.<Extension>
```

LoincVersion and ArtifactVersion would use the Semantic Versioning style to define Major, Minor, and Patch versions (and “prerelease” labels).

Examples

```
LOINC_2.62_Access.zip  
LOINC_2.59.1_PanelsAndForms_1.0.zip  
LOINC_2.62_GroupFile_1.0_Alpha_2.zip
```

LOINC Release Versions

Overall, the LOINC Release Version is primarily related to contents and format of the LoincTable and LoincTableCore (and a few other files bundled with them).

This includes both CSV and Access files of the LoincTable, but not, for example, the RELMA program or other Accessory Files.

LOINC Versioning

Major

A structural change to the LoincTableCore.csv file

Minor

Edits to existing contents of fields in the LoincTableCore

New LOINC terms added (to LoincTableCore and LoincTable)

Structural changes to LoincTable

Patch

Edits to existing contents of the fields ONLY in LoincTable

(changes to contents in LoincTableCore would be a Minor version change)

Artifact Release Versions

Many release artifacts (Top2000, Groups, Mapping files) have a “life of their own”.

Their structure can evolve independently of the underlying LOINC version.

We’d like to signal artifact changes separately from main LOINC changes.

The “significance” of changes may vary by artifact, so may need to define these differently by artifact.

Artifact Versioning

Major

A structural change the release artifact format

Minor

Edits to existing contents of fields

New records added

Patch

Edits to existing contents of the fields ONLY

(additions to rows contents would be a Minor version change)

Pre-release VersionType and Increment

A pre-release version MAY be denoted by appending a hyphen and a series of dot separated identifiers immediately following the patch version. (e.g. alpha.1)

Alternate Proposal for Artifacts

Should *ANY* structural change trigger a Major change?

Perhaps additions of fields need not trigger a Major change.

LOINC License

Propose using versioning here too (License.txt file)
Could be very helpful to users.

Major

Changes to other clauses, attributions needed, etc

Minor

New artifacts added

New external required attributions added

Patch

updates to copyright end year only

Open discussion

general strategy

rollout-plan