NETMOD YANG Semver Update

NETMOD WG
Jul 2022

Presenting on behalf of the weekly versioning call attendees:
Joe Clarke

IETF 114
Current Status

• Released -07 with some WG LC comments addressed:
  • Use SemVer to denote the reference the SemVer 2.0.0 spec
  • Clarify use of "semantic version" as a concept as opposed to referencing SemVer where we refer to YANG semver specifically
  • Fix a prefix typo
  • Compress acknowledgments
Outstanding Issues

• From a YANG Semver, cannot tell what derives from what
  • Revision-labels are not linear
• It’s SemVer but it isn’t
• YANG Semver is trying to encode the release train
• Too much work/noise to bump YANG Semver for WIP
Cannot Tell What Derives From What

Can we understand the development sequence of module versions from the YANG-Semver revision-label?

**NO!**

The same set of revision-labels might be a result of different development paths.

Just as the same set of revision dates may be a result of different development paths.

**Branching, not labeling, destroys the simple sequence.**

Read the sequence of revision statements in the YANG module to understand the development path.
It’s SemVer But It Isn’t

• YANG Semver is 100% compatible with SemVer 2.0.0
• YANG Semver is a super-set
• One can use pure SemVer rules in their revision-label statements and be fully compatible with YANG Semver
• TODO: Clarify in text what is SemVer 2.0.0 and what is unique to YANG Semver (Issue #140)
YANG Semver Encodes The Release Train

• This is not a goal of YANG Semver and so this is not necessarily true
• Vendors *do* have the flexibility to use `revision-label` values that truly encode the release train
• YANG Semver is only a means to signal changes
Bumping YANG Semver For WIP Modules

• Modules, even in draft state, MUST have unique YANG Semver values (if they use YANG Semver as a revision-label scheme)
• If the module does not change between draft revisions, no change to its revision-label value is required
• Changing revision-label values helps when a vendor chooses to implement a pre-ratified version of a module
• TODO: Add more clarity to when to change revision-label (Issue #142)