NETMOD YANG Module Versioning Update

NETMOD WG
March 2024

Presenting on behalf of the weekly versioning call attendees:
Robert Wilton (who might be hungover)

IETF 119
Recap

Since IETF118, many changes were made to YANG Module Versioning. Those changes were based on discussions and rough consensus in the following forums

1) WG meeting at IETF 118
2) Mailing list e.g. https://mailarchive.ietf.org/arch/msg/netmod/yT7uVKLNs6UAH3IC_M1QUcbkyek/
3) Weekly YANG Versioning meetings
4) Hedgedoc polls
Main changes made since IETF118
Removed: extensions \textit{revision-label(–scheme)}

• As presented at IETF118, based on WG LC discussions, we have gone back to a single versioning scheme for YANG modules. Hence, the \textit{revision-label-scheme} extension has been removed

• Given that a single versioning scheme is sufficient, the \textit{revision-label} extension has been moved to the YANG Semver document (where it has been renamed to \textit{version})
Removed: resolving ambiguous imports

As presented at IETF118, this has been removed from the document. Note, RFC 7950, section 5.6.5, paragraph 5 (below) does consistently define how to build the schema. The change in the document was to always prioritize an implemented module over the most recent implemented *or* import-only revision. This could be considered for YANG-Next.

If a server lists a module C in the
"/modules-state/module" list from "ietf-yang-library"
and there are other modules Ms listed that import C
without specifying the revision date of module C, the
server MUST use the definitions from the most recent
revision of C listed for modules Ms.
Extension *recommended-min-date* (import statement)

- Extension *recommended-min* has been replaced by *recommended-min-date*. As the name suggests, the recommended minimum version is date specific.
- Only a **date comparison** is made when deciding whether a specific revision abides by the *recommended-min-date* date argument.
- This extension works well for linear revision history, but not for branched revision history. The YANG Semver document has a similar extension for branched revision history.
Extension *recommended-min-date* (contd)

This example recommends module revisions for import whose revision date is or comes after 2019-04-01. It includes the following revisions: 2019-04-01, 2019-05-01 and 2019-06-01, *even though revision 2019-05-01 may not contain what is desired from 2019-04-01.*

```
Module revision date
2019-01-01
  |
2019-02-01
  |
2019-03-01
  |
      \ 2019-04-01
  |
2019-05-01
  |
      2019-06-01
```

import example-module {
    rev:recommended-min-date 2019-04-01;
}

RFC7950 section 5.2 has the following filename format: 

my-module@<revision-date>.yang

To facilitate “importing by revision”, this document proposed an alternate/additional filename format: 

my-module#<revision-label>.yang

We could not reach consensus on this new filename format

Per Andersson tested pyang fix for this in the 118 hackathon

Could be reconsidered as part of YANG next
Augment of ietf-yang-library

• Advertising how deprecated & obsolete nodes are handled
• Renamed module ietf-yang-status-conformance augments ietf-yang-library (note: rev labels moved to YANG semver)

```
module: ietf-yang-status-conformance
  augment /yanglib:yang-library/yanglib:schema:
    +--ro deprecated-nodes-implemented? boolean
    +--ro obsolete-nodes-absent?     boolean
```
Various Other Changes

• Removed versioning of YANG instance data. Rough consensus to defer this to a new version of the YANG instance data document

• Removed text which states that all whitespace changes (including whitespace between statements) require a new revision to be published
Current contents

• New extension “non-backwards-compatible” to document when NBC changes have occurred

• New extension “recommended-min-date” which extends the import statement with a minimum revision suggestion to help document inter-module dependencies

• Guidelines for managing the lifecycle of YANG modules

• Augments ietf-yang-library to report how "deprecated" and "obsolete" nodes are handled by a server
Next Steps

• Plan to request another WGLC after IETF 119