NETMOD YANG Versioning Solution Update

NETMOD WG
July 2022

Presenting on behalf of the weekly versioning call attendees:
Tom Hill

IETF 114
Agenda

This presentation:
   Solution Overview – Tom Hill

Following presentations on individual drafts:
   Module Versioning – Balazs Lengyel
   YANG SemVer – Joe Clarke

See backup slides for solution overview & summary of Module Versioning and YANG Semver drafts.
Recap - YANG Versioning Solution Overview

Complete solution consists of five drafts:

1. Updated YANG Module Revision Handling:
   Notify nbc changes between module revisions, allows branched revision history, revision-labels

2. Module semantic version number scheme:
   YANG semver for module revision-labels and package versioning

3. YANG schema comparison tooling:
   Tooling to algorithmically compare module or schema revisions

4. Versioned YANG packages:
   Versioning at the schema level rather than individual modules

5. Protocol operations for package version selection:
   Devices can support multiple schema versions, clients can select for session

Latest working drafts can all be found here: https://github.com/netmod-wg/yang-ver-dt
Recap - YANG Versioning Solution Overview

Links for published drafts:

https://datatracker.ietf.org/doc/draft-ietf-netmod-yang-schema-comparison
General update on weekly Versioning calls

• Authors + interested parties continue to meet on a weekly call to progress this work
  • Meetings are open to all
  • Regular participation from people across 5+ different companies (mostly equip. vendors, participation from more operators/users would be beneficial)
  • Key issues are brought back to WG mailing list
  • Weekly meeting is currently on Tues @2pm UK time / 9am Eastern. Thanks to the authors and contributors for their regular attendance

General update on weekly Versioning calls

Focus of weekly meetings since IETF113:

1) Processing feedback from WG Last Call on Module Versioning and YANG Semver drafts
   • a number of complex issues raised requiring significant discussion and debate
   • in process of posting responses back to the mailing list
   • some updates to the drafts will be required

2) Briefly started looking at Schema Comparison draft again
   • LC on Module Versioning draft made us realize the schema comparison is fairly integral
   • May move some parts of Schema Comparison into Module Versioning (e.g. per element NBC/BC tags)

We didn’t look at Packages or Version Selection drafts.
Next Steps

1) Finish processing all comments received from WG LC for Module Versioning and YANG Semver:
   • respond and discuss on mailing list
   • drive remaining github issues to closure
   • updates to the drafts

2) Weekly call group recommends doing another WG LC for Module Versioning and YANG Semver with intention to bring them to RFC:
   • Don’t wait any longer for remaining 3 drafts (Comparison, Packages, Selection)
   • Work has been taking a long time to converge
   • O-RAN is asking to have these 2 drafts completed

3) Phase work on remaining documents: Schema comparison, Packages, Version selection
BACKUP SLIDES
Module Revision Handling + YANG Semver Drafts
Summary of main requirements & recommendations

draft-ietf-netmod-yang-module-versioning - Updated YANG Module Revision Handling
draft-ietf-netmod-yang-semver - YANG Semantic Versioning

A) Mark a revision as non-backwards-compatible

module example-module {
  namespace "urn:example:module";
  prefix "prefix-name";

  import ietf-yang-revisions { prefix "rev"; }
  import ietf-yang-semver { prefix "ysver"; }

  rev:revision-label-scheme "ysver:yang-semver";

  revision 2019-06-01 {
    rev:revision-label 3.1.0;
    description "Add new functionality.";
  }

  revision 2019-03-01 {
    rev:revision-label 3.0.0;
    rev:non-backwards-compatible;
    description "Add new functionality. Remove some deprecated nodes.";
  }

  revision 2019-02-01 {
    rev:revision-label 2.0.0;
    rev:non-backwards-compatible;
    description "Apply bugfix to pattern statement";
  }
}
Module Revision Handling + YANG Semver Drafts
Summary of main requirements & recommendations

B) Import by revision-or-derived

Specify a *minimum* revision for import, but any descendant is also conformant.

Use the latest descendant available.

```
import example-module {
    rev:revision-or-derived 3.0.0
}
```
Summary of main requirements & recommendations

C) YANG Semver revision-label

X.Y.Z = Major.Minor.Editorial

Major -> change for NBC changes
Minor -> additional leaves, etc
Editorial -> fix a description

Example YANG module with branched revision history.

<table>
<thead>
<tr>
<th>Module revision date</th>
<th>Revision label</th>
</tr>
</thead>
<tbody>
<tr>
<td>2019-01-01</td>
<td>&lt;= 1.0.0</td>
</tr>
<tr>
<td>2019-02-01</td>
<td>&lt;= 2.0.0</td>
</tr>
<tr>
<td>2019-03-01</td>
<td>&lt;= 3.0.0</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>2019-04-01</td>
<td>&lt;= 2.1.0</td>
</tr>
<tr>
<td>2019-05-01</td>
<td>&lt;= 2.2.0</td>
</tr>
<tr>
<td>2019-06-01</td>
<td>&lt;= 3.1.0</td>
</tr>
</tbody>
</table>

Optional compatibility extension:
rev:revision-label 3.0.2_non_compatible
Module Revision Handling + YANG Semver Drafts
Summary of main requirements & recommendations

D) Filename recommendation

my-module#3.2.0.yang