Semantic Versioning for YANG Artifacts

draft-verdt-netmod-yang-semver

NETMOD WG
November 19, 2019

Netmod YANG Version Design Team
Joe Clarke (presenting)
What’s The Same Since -00

• Modified semver syntax and rules
• Notions of NBC changes, BC changes, and editorial changes
• Modules can have a semantic version associated with them
Modified Semver (Recap)

- x.y.zm|M
  - x major version, stepped for NBC changes
  - y minor version, stepped for BC changes
  - z patch version stepped for editorial changes
  - m used for BC changes, needed when next “y” has already been used, and the old version MUST be updated
  - M used for NBC changes, needed when next “x” has already been used, and the old version MUST be updated

- The same x.y.z triplet MUST NOT be reused (independent of m|M)

- m|M optional

- Beta versions of the form 0.y.z do not follow these rules
YANG Semver Example
(Recap)

YANG semantic version numbers for an example module:

0.1.0

0.2.0

1.0.0

1.1.0 -> 1.1.1m -> 1.1.2M

1.2.0 -> 1.2.1M -> 1.2.2M

1.3.0 -> 1.3.1

2.0.0

3.0.0

3.1.0

0.1.0 - first beta module version
0.2.0 - second beta module version (with NBC changes)
1.0.0 - first release (may have NBC changes from 0.2.0)
1.1.0 - added new functionality, leaf "foo" (BC)
1.2.0 - added new functionality, leaf "baz" (BC)
1.3.0 - improve existing functionality, added leaf "foo-64" (BC)
1.3.1 - improve description wording for "foo-64" (Editorial)
1.1.1m - backport "foo-64" leaf to 1.1.x (BC)
1.2.1M - NBC bug fix, not required in 2.0.0 due to model changes (NBC)
1.3.0 - introduce new leaf "wobble" (BC)
2.0.0 - change existing model for performance reasons, e.g. re-key list (NBC)
3.0.0 - NBC bugfix, rename "baz" to "bar"; also add new BC leaf "wibble"; (NBC)
1.2.2M - backport NBC fix, changing "baz" to "bar"
1.2.1M - backport NBC fix, changing "baz" to "bar"
1.2.2M - backport "wibble". This is a BC change but "M" modifier is sticky.
What’s New Since -00

• The term YANG Artifact has been introduced to extend semver to packages as well as modules
• The semver construct is documented to be a revision-label value
• No longer updates 7950
• Full support has been added for semver 2.0.0
• The semver typedef is formally defined, and rules re-stated that a revision-label that follows this pattern should be treated with modified semver rules
Semver As Revision-label

OLD:

```
revision 2018-02-28 {
    description "Added leaf 'wobble'";
    semver:module-version "3.1.0";
    reference "product-description ACME-Router v12";
}
```

NEW:

```
import ietf-yang-revisions {
    prefix rev;
}

revision 2018-02-28 {
    description "Added leaf 'wobble'";
    rev:revision-label "3.1.0";
    reference "product-description ACME-Router v12";
}
Support For Full Semver 2.0.0

- Semver 2.0.0 defines optional pre-release and build metadata to follow a version string
- E.g., Pre-release: 1.0.0-alpha, 1.0.0M-alpha.1, 1.0.0-0.3.7, 1.0.0-x.7.z.92
- E.g., Build: 1.0.0m-alpha+001, 1.0.0+20130313144700, 1.0.0-beta+exp.sha.5114f85
- This MD is ignored for purposes of YANG semver (this just allows us to be completely compatible with semver 2.0.0)
YANG Semver Typedef

typedef version {
    type string {
        pattern '\d+.\d+.\d+[mM]?(-[\w/d.]+)?([+][-]+)?';
    }
    description
        "Represents a YANG semantic version number. Note:
        additional rules apply to the dot-separated numeric identifiers
        which are spelled out in the reference for this typedef.";
    reference
        "RFC XXXX: YANG Semantic Versioning.";
}