Network Working Group
Internet-Draft
Intended status: Informational
Expires: May 6, 2020

R. Wilton, Ed.
Cisco Systems, Inc.

November 3, 2019

YANG Versioning Solution Overview
draft-verdt-netmod-yang-solutions-02

Abstract

This document gives a brief overview of the different drafts that comprise a full solution to the YANG versioning requirements draft. The purpose of this draft is to help readers understand how the discrete parts of the YANG versioning solution fit together during working group development of the solution drafts.

Status of This Memo

This Internet-Draft will expire on May 6, 2020.

Copyright Notice

Table of Contents

1. Introduction 2
2. Solution Drafts 2
  2.1. Updated YANG Module Revision Handling 3
  2.2. Module semantic version number scheme 4
  2.3. Versioned YANG packages 4
  2.4. Protocol operations for package version selection 5
  2.5. YANG schema comparison tooling 5
3. Contributors 6
4. Security Considerations 6
5. IANA Considerations 7
6. References 7
  6.1. Normative References 7
  6.2. Informative References 7
Author's Address 7

1. Introduction

[I-D.ietf-netmod-yang-versioning-reqs] documents the requirements for any solution to the YANG [RFC7950] versioning problem. Chapter 5 lists the formal requirements that a complete solution requires.

The aim of this draft is to help readers understand how the different solution drafts fit together, and also which drafts contribute solutions to particular individual requirements. The overall solution comprises five individual drafts:

1. [I-D.verdt-netmod-yang-module-versioning]

2. [I-D.verdt-netmod-yang-semver]

3. [I-D.rwilton-netmod-yang-packages]

4. [I-D.wilton-netmod-yang-ver-selection]

5. YANG schema comparison tooling (not yet published)

Open issues, across all of the solution drafts are tracked at <https://github.com/netmod-wg/yang-ver-dt/issues>.

2. Solution Drafts

The complete solution to the YANG versioning requirements comprises five solution drafts, that are summarized below.

2.1. Updated YANG Module Revision Handling

In summary, [I-D.verdt-netmod-yang-module-versioning] specifies minimal extensions and updates to the YANG language, YANG Library, and YANG author guidelines to provide more flexible YANG module revision handling. The intent is that these changes and extensions could be folded into future revisions of the updated specifications. The draft provides a solution for all requirements except Req 2.2, Req 3.1 and Req 3.2.

The extensions and changes in the draft can be summarized thus:

- It defines a YANG extension statement to indicate where non-backwards-compatible changes have occurred in a module's revision history.

- It relaxes the rules for the module revision history to allow for a non-linear module revision history. I.e., any given module revision may have multiple revisions directly derived from it.

- It defines a new import extension statement that restricts the allowed module revisions that satisfy the import to only those derived from a specified module revision.

- It defines a revision label extension statement to allow an informative name to be associated with a particular revision date, and to be used in import statements, YANG module filenames, and is available in YANG library. One example of how the revision label could be used is to associate a semantic versioning scheme to YANG module revisions.

- It updates the YANG rules for changes between module revisions that are allowed to be classified as backwards-compatible. In particular, marking a node as obsolete is no longer classified as a backwards compatible change.

- It provides updated guidance on how servers handle deprecated and obsolete YANG nodes and augments YANG library with additional leaves to report the server's behavior to clients.

- It provides an extension statement to allow a description statement to be associated with a YANG status statement, providing more information about why the status has changed.

- It defines how versioning relates to YANG instance data.

- It refines the guidelines for updating modules, taking into consideration that non-backwards-compatible changes are sometimes necessary for various pragmatic reasons.

2.2. Module semantic version number scheme

[I-D.verdt-netmod-yang-semver] defines a semantic versioning scheme derived from the semver.org 2.0.0 specification that can be used in conjunction with the revision label extension statement defined in Section 2.1 to allow semantic version numbers to be used to manage the revision lifecycle of YANG modules. This draft provides an enhanced solution for Req 2.1, but organizations authoring modules are not obliged to use this specific versioning scheme, and could choose a different overlaid versioning scheme, or none at all and rely solely on revision dates.

The aims of the YANG semantic versioning scheme are:

To generally allow clients to determine whether NBC changes have occurred between two revisions from the version number alone, without having to check the full revision history.

To give a more informative identifier for a branched revision history over revision dates alone.

To allow revision branches that contain fixes for published non-latest releases.

2.3. Versioned YANG packages

The two previous drafts address version and revision management of individual modules. [I-D.rwilton-netmod-yang-packages] provides a mechanism to group a set of related YANG modules revisions together, into a construct called a YANG package, and to apply a version scheme to the group.

The core part of this draft are YANG module definitions that define a YANG package, that are used as an augmentation to YANG library, and also in YANG instance data documents for offline access.

The principle aims of the packages draft are:

To provide an alternative simpler mechanism to manage conformance of modules. Rather than checking conformance against a set of individual YANG module revisions, it should be easier to check for conformance against a much smaller set of YANG package versions.

To provide an easier mechanism for clients to check conformance with a server. Rather that downloading and comparing all individual module revisions, the client can just check whether the package version is compatible. The package definition could be retrieved and cached from multiple sources.

The YANG packages draft does not address any of the versioning requirements directly, but provides the foundation building blocks for the version selection solution, described in Section 2.4, that addresses Reqs 3.1 and 3.2.

2.4. Protocol operations for package version selection

[I-D.wilton-netmod-yang-ver-selection] specifies a solution for requirements 3.1 and 3.2 via the use of [I-D.rwilton-netmod-yang-packages] and a protocol based version selection scheme that can be used by clients to choose a particular YANG datastore schema from the set of datastore schema that are supported by the server.

The version selection optionally allows:

Servers to support a single, selectable YANG package at a particular version, that is used for all NETCONF/RESTCONF interactions.

Servers to support multiple selectable YANG packages and package versions, with different clients able to concurrently access different packages and different package versions.

2.5. YANG schema comparison tooling

A tooling based solution is proposed for Req 2.2, that allows two YANG schema versions to be algorithmically compared, with the algorithm reporting the list of differences between the two YANG schema and whether each change is regarded as being backwards-compatible, or non-backwards-compatible. Annotations to the YANG modules, via the use of extension statements, may help improve the accuracy of the comparison algorithm, particularly for statements that are very hard to algorithmically classify the scope of any differences (e.g., a change in the semantic behaviour of a data node defined via modifications to the associated YANG description statement). Given that Req 2.2 is a softer requirement, and practical experience with the tooling is required, it is proposed that this work is deferred at this time.

When comparing a module schema, a tool would also be able to take into account enabled features, deviations, and the subset of the

schema being used by the client. This would allow a tooling based approach to give a more accurate answer as to whether a client would be affected when upgrading between two software versions, than looking at the revision history, or comparing semantic version numbers.

3. Contributors

This document grew out of the YANG module versioning design team that started after IETF 101. The following individuals are (or have been) members of that design team and have contributed to defining the problem, specifying the requirements, and working on a solution:

- Balazs Lengyel
- Benoit Claise
- Ebben Aries
- Jason Sterne
- Joe Clarke
- Juergen Schoenwaelder
- Mahesh Jethanandani
- Michael (Wangzitao)
- Qin Wu
- Reshad Rahman
- Rob Wilton
- Susan Hares
- Wu Bo

4. Security Considerations

The document does not define any new protocol or data model. There is no security impact.

5. IANA Considerations

None

6. References

6.1. Normative References

[I-D.ietf-netmod-yang-versioning-reqs]
Clarke, J., "YANG Module Versioning Requirements", draft-ietf-netmod-yang-versioning-reqs-01 (work in progress), July 2019.

[RFC7950] Bjorklund, M., Ed., "The YANG 1.1 Data Modeling Language", RFC 7950, DOI 10.17487/RFC7950, August 2016, <https://www.rfc-editor.org/info/rfc7950>.

6.2. Informative References

[I-D.rwilton-netmod-yang-packages]
Wilton, R., "YANG Packages", draft-rwilton-netmod-yang-packages-02 (work in progress), October 2019.

[I-D.verdt-netmod-yang-module-versioning]
Claise, B., Clarke, J., Rahman, R., Wilton, R., Lengyel, B., Sterne, J., and K. D'Souza, "Updated YANG Module Revision Handling", draft-verdt-netmod-yang-module-versioning-01 (work in progress), October 2019.

[I-D.verdt-netmod-yang-semver]
Claise, B., Clarke, J., Rahman, R., Wilton, R., Lengyel, B., Sterne, J., and K. D'Souza, "YANG Semantic Versioning", draft-verdt-netmod-yang-semver-01 (work in progress), October 2019.

[I-D.wilton-netmod-yang-ver-selection]
Wilton, R., Rahman, R., and J. Clarke, "YANG Schema Version Selection", draft-wilton-netmod-yang-ver-selection-01 (work in progress), November 2019.

Author's Address

Robert Wilton (editor)
Cisco Systems, Inc.

Email: rwilton@cisco.com