

Summary of IETF NETMOD work on YANG Versioning and Packaging



Introduction



- YANG Versioning and Packaging have come a long way since the “revision” statement was introduced in YANG 1.0.
- This presentation describes some of the features related to revision, module tagging, semantic versioning, change compatibility
- Some of this IETF work has reached RFC level and some is work in progress

- Features Discussed
 - YANG 1.1 supported features (RFC 7950)
 - Revision statement
 - Deviation statement
 - YANG Library (RFC 8525)
 - YANG Module Tags (RFC 8819)
 - Drafts related to:
 - YANG Module Versioning
 - YANG Semantic Versioning
 - YANG Packages
 - YANG Version Selection
 - YANG Schema Comparison

RFC 7950 and RFC 8407



- <https://tools.ietf.org/html/rfc7950#section-7.1.9> describes the “revision” statement
 - *The “revision” statement specifies the editorial revision history of the module, including the initial revision. A series of “revision” statements detail the changes in the module’s definition. The argument is a date string in the format “YYYY-MM-DD”...*
- <https://tools.ietf.org/html/rfc7950#section-5.1.1> describes how to import and include by revision
- <https://tools.ietf.org/html/rfc7950#section-5.6.3> describes how to support deviations in a server implementation
- <https://tools.ietf.org/html/rfc8407#section-4.7> and <https://tools.ietf.org/html/rfc8407#section-4.8> provide guidance on the lifecycle associated with YANG and the use of revision statements.

RFC 8525 YANG Library



- <https://tools.ietf.org/html/rfc8525>
 - *This document describes a YANG library that provides information about the YANG modules, datastores, and datastore schemas used by a network management server.*
- Provides a model that can be used by a YANG client application to retrieve
 - name
 - revision
 - submodule list
 - feature list
 - deviation list
- From a YANG server implementation

RFC 8819 YANG Module Tags



- <https://tools.ietf.org/html/rfc8819>
 - *This document provides for the association of tags with YANG modules. The expectation is for such tags to be used to help classify and organize modules. A method for defining, reading, and writing modules tags is provided.*
- <https://tools.ietf.org/html/rfc8819#section-1.1> describes the potential use cases for YANG Module Tags
- <https://tools.ietf.org/html/rfc8819#section-7.1> process for IANA

- IEEE 802.1 impact
 - The tagging feature could be used to group all related modules based on functionality provided.
 - For example there could be the following tags:
 - ieee802:time-sensitive-networking
 - ieee802:provider-bridging
 - ieee802:connectivity-fault-management
 - ieee802:ethernet
 - ieee1588:time-synchronization

Versioning Solution Overview



- <https://github.com/netmod-wg/yang-ver-dt/blob/master/yang-solution-overview/draft-ietf-netmod-yang-solutions.txt>
 - Provides an overview of how the current work on versioning in the NETMOD WG is divided up.
- <https://datatracker.ietf.org/meeting/110/materials/slides-110-netmod-yang-versioning-update-00>
 - Provides the status of the work from IETF 110
 - Summary:
 - Updated YANG Module Revision Handling
 - Can notify of nbc (non-backward compatible) changes between module revisions, allows branched revision history, revision-labels
 - Module semantic version number scheme
 - Allows use of YANG semver for module revision-labels and package versioning
 - Versioned YANG packages
 - Versioning at the schema level rather than individual modules
 - Protocol operations for package version selection
 - Devices can support multiple schema versions, clients can select for session
 - YANG schema comparison tooling
 - Tooling to algorithmically compare module or schema revisions

YANG Semantic Versioning



- <https://github.com/netmod-wg/yang-ver-dt/blob/master/yang-semver/draft-ietf-netmod-yang-semver.txt>
 - *This document specifies a scheme and guidelines for applying a modified set of semantic versioning rules to revisions of YANG modules. Additionally, this document defines a revision-label for this modified semver scheme.*
- <https://datatracker.ietf.org/meeting/110/materials/slides-110-netmod-yang-semver-00>
 - Presentation from IETF 110

YANG Module Versioning



- <https://github.com/netmod-wg/yang-ver-dt/blob/master/yang-module-versioning/draft-ietf-netmod-yang-module-versioning.txt>
 - *This document specifies a new YANG module update procedure that can document when non-backwards-compatible changes have occurred during the evolution of a YANG module.*
- <https://datatracker.ietf.org/meeting/110/materials/slides-110-netmod-module-versioning-update-open-issues-00>
 - Status presentation from IETF 110

YANG Packages



- <https://tools.ietf.org/html/draft-ietf-netmod-yang-packages-01>
 - *This document defines YANG packages, a versioned organizational structure holding a set of related YANG modules that collectively define a YANG schema. It describes how packages: are represented on a server, can be defined in offline YANG instance data files, and can be used to define the schema associated with YANG instance data files.*
- This document pulls together a practical way to treat groups of YANG modules as a package.
- <https://tools.ietf.org/html/draft-ietf-netmod-yang-packages-01#section-3> provides useful background and rationale for the current work.

Important Terminology



- <https://tools.ietf.org/html/rfc7950#section-3>
- Model
 - A collection of modules
 - YANG structures data models into modules and submodules. (<https://tools.ietf.org/html/rfc7950#section-4.1>)
- Module
 - Defines hierarchies of schema nodes
- Schema Node
 - A node in a schema tree. (action, container, leaf, leaf-list, list, choice, case, rpc, input, output, notification, anydata, and anyxml)
- Schema tree
 - The definition hierarchy specified within a module

Documents



- [RFC 7950 The YANG 1.1 Data Modeling Language](#)
- [RFC 8407 Guidelines for Authors and Reviewers of Documents Containing YANG Data Models](#)
- [RFC 8525 YANG Library](#)
- [RFC 8819 YANG Module Tags](#)
- [YANG Versioning Drafts \(Github\)](#)
 - [YANG Versioning Solution Overview Draft](#)

