YANG Schema Comparison

draft-ietf-netmod-yang-schema-comparison-06

IETF 125 NETMOD WG 2026-03-18

Per Andersson <per.ietf@ionio.se>
Rob Wilton <rwilton@cisco.com>
Michal Vasko <mvasko@cesnet.cz>

Intro

- Algorithm for comparing two revisions of a YANG module
- Determine scope of changes and list changes between revisions
  - BC and NBC changes
- Define a YANG module describing output of the comparison algorithm
- Algorithm output can help select appropriate revision-label or YANG Semver

Solution and Terminology

- ietf-yang-schema-comparison
  - YANG module defining comparison output
- Parsed schema
  - The actual YANG source
- Compiled schema
  - Data on-the-wire

Changes since IETF 124

- Cleared all issues
- Document updates
  - Add text for compiled/parsed schema changes
  - YANG module updated
  - Updated example
  - Terminology updated
  - Editorial changes
- libyang implementation updates
  - Reference implementation completed

Compiled/Parsed schema changes

- Changes generated either
  1) Only compiled
  2) Both compiled and parsed
- Method and report
  - Signaled via if-feature in ietf-yang-schema-comparison
  - For 2)
    - Compiled changes generated
    - Then all other in parsed form (skipping compiled)
    - Changes only reported once (not in both compiled and parsed)

YANG module updates, general

- Support for all parsed schema statements
- Schema tree structure unified
- Clarified adding “backwards-compatible” extension to an instance, it will be true for all substatements

YANG module updates, details #1

- Introduced schema-stmt-path
- typedef stmt-type enum
  - Added deviate, deviation, extension, feature, if-feature, import, include
- “node” enum
  - Can also include: input, output, augment, grouping
  - Added prefix, refine, revision-date, typedef, yang-version
- Added ext-instance-stmt
  - Describes chages to an extension instance

YANG module updates, details #2

- Grouping refine-define-node-substmts
  - Renamed from refine-node-substmts
  - Includes deviations
- Grouping node-substmts
  - Includes all substatements of a data-definition node
  - Added when-stmts, status-stmt
  - Added leaf-list with nodes referenced by key
  - Added leaf ordered-by: enum indicating ordered-by system or user

YANG module updates, details #3

- Grouping module-substmts
  - Extended to includes all substatements to a module definition
  - Lists changed to presence containers
    - Extension, feature, identity, deviation
    - Only exist if there is a difference
  - Uses ext-instance-stmts

YANG module updates, details #4

- Added grouping type-parsed-substmts
  - Describes a parsed type statement change
- Added grouping parsed-substmts
  - All parsed-only substatements that can be direct substatements of a module or nested in a node
- Added parsed-info
  - Information about all the changed parsed-only statements

Updated example

- Compiled Module example changed into Relaxed Range Module example
  - Diff showing both compiled and parsed changes

Terminology updated

- Added terminology from RFC 7950
  - module, data node, data definition statement
- Referencing terminology from draft-ietf-yang-versioning-reqs
  - Backwards-compatible, Non-backwards-compatible

Implementation

- libyang implementation completed
  - Supports ietf-yang-schema-comparison
  - Supports both compiled and parsed schema changes

Next steps

- Testing and feedback by users
  - Ideally with their custom YANG modules
- Evaluate Joe’s pyang schema comparison algorithm
  - Can detect editorial changes
  - Could it be incorporated and improve algorithms proposed?
- Investigate interaction with YANG Packages
- Add more examples
- Final editorial touches
- WGLC

Thank you!

Questions?