

IETF YANG Versioning Packages Update

draft-ietf-netmod-yang-packages-07

NETMOD WG IETF 125, Mar 2025

Presenting (on behalf of authors): Rob Wilton

Main contributors: Jason Sterne, Reshad Rahman, Joe Clarke

Solution Overview

Solution Overview draft: [draft-ietf-netmod-yang-versioning-reqs-11](#)

Proposed full solution is made up of 6 drafts (at different levels of maturity):

1. Updated YANG Module Revision Handling ([draft-ietf-netmod-yang-module-versioning-15](#))
2. Module semantic version number scheme ([draft-ietf-netmod-yang-semver-24](#))
3. Yang module filename convention ([draft-ietf-netmod-yang-module-filename-03](#))

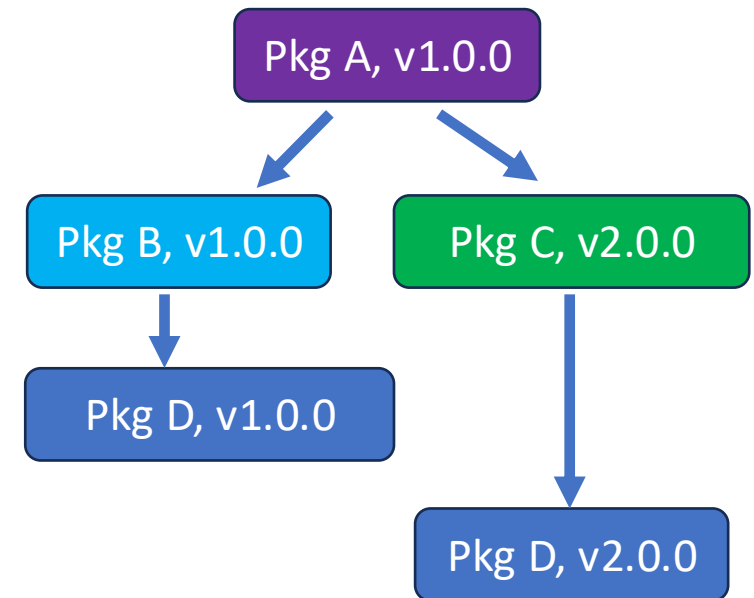
4. Guidance for IANA and RFC editor ([draft-ietf-netmod-iana-yang-guidance-00](#))
5. **Versioned YANG packages** ([draft-ietf-netmod-yang-packages-06](#))
6. YANG schema comparison tooling ([draft-ietf-netmod-yang-schema-comparison-05](#))
7. Protocol operations for package version selection ([draft-ietf-netmod-yang-ver-selection-00](#))



**Potentially 4b: IETF
Packages draft**

YANG Packages Recap 101

- YANG Packages are an alternative mechanism for managing sets of YANG modules, features and mount
- They can be defined hierarchically
- A package logically resolves into a YANG schema – like the information provided by YANG library
- Packages are versioned, using YANG Semver
- Packages should be made available offline and referenced by a URL
- Tooling to allow package versions to be compared/queried



Summary

- Working through open issues
- Closed/resolved 28 issues since IETF 124
- 25 issues that are open
 - 5 are ready for last checks before closing them
 - Good progress made on quite a few other others
 - I.e., less work remaining than issue count implies
- YANG package model seems to be stabilizing
- Somewhat hoping to be ready for IETF LC before IETF 126
 - May still be a bit to far out

Changes between -06 and -07 (1)

- Improvements to terminology
 - A bit more cleanup still to do
- Refinements to package definition rules:
 - No circular package imports
 - Added guidance on invalid package definitions (from tooling experience)
 - Allowed redundant information
- More guidance on package usage
 - SDO/Industry defined packages vs vendor/device defined packages
 - Updated considerations for listing implemented vs import-only modules

Changes between -06 and -07 (2)

Refinements to the Packages Data Model:

- Small tweaks:
 - Defined Module-name type, moved features, plurals -> singular for lists
- Import-only-modules list:
 - Changed “replaces-version” to explicit list under excludes
- Mounts:
 - Replaced “replaces-packages” leaf with a single per mount “inherit-packages” leaf

Packages tree diagram

```
grouping yang-pkg-instance:
  +-- name          pkg-name
  +-- version       pkg-version
  +-- timestamp?   yang:date-and-time
  +-- organization? string
  +-- contact?     string
  +-- description? string
  +-- reference?   string
  +-- complete?    boolean
  +-- includes
  |   +-- package* [name]
  |   |   +-- name          pkg-name
  |   |   +-- version       pkg-version
  |   |   +-- location*    inet:uri
  |   +-- module* [name]
  |   |   +-- name          module-name
  |   |   +-- version       version-or-rev-date
  |   |   +-- location*    inet:uri
  |   |   +-- submodule* [name]
  |   |   |   +-- name          module-name
  |   |   |   +-- version       version-or-rev-date
  |   |   |   +-- location*    inet:uri
```

```
+-- import-only-module* [name version]
  |   +-- name          module-name
  |   +-- version       version-or-rev-date
  |   +-- location*    inet:uri
  |   +-- submodule* [name]
  |   |   +-- name          module-name
  |   |   +-- version       version-or-rev-date
  |   |   +-- location*    inet:uri
  |   +-- feature*     scoped-feature
+-- excludes
  |   +-- module*       module-name
  |   +-- import-only-module* [name]
  |   |   +-- name          module-name
  |   |   +-- version*     version-or-rev-date
  |   +-- feature*     scoped-feature
+-- mount* [mount-path]
  +-- mount-path       mount-yxpath
  +-- inherit-packages? boolean
  +-- package* [name]
  |   +-- name          pkg-name
  |   +-- version       pkg-version
  |   +-- location*    inet:uri
  +-- parent-reference* mount-yxpath
```

Changes between -06 and -07 (3)

- *Reminder:*
 - *YANG Packages can define alternative schema to YANG library. I.e., using packages rather than module-sets.*
- Added **supported-features** leaf-list on datastore schema:
 - To allow a device to dynamically change enabled/disabled features without changing the published packages

```
module: ietf-yl-packages
```

```
augment /yanglib:yang-library/yanglib:schema:  
  +--ro package* [name version]  
  | +--ro name      -> /pkgs:packages/package/name  
  | +--ro version   leafref  
  +--ro supported-feature*   pkg-types:scoped-feature
```

Changes between -06 and -07 (4)

- Defines .ypkg file type for YANG package files
- Reworked/Improved text on Package Conformance
 - New section of package comparison tools
- Updated security considerations
- IANA Considerations:
 - Now defines a “YANG Package Registry” like IANA’s “YANG Module Registry”
 - Now defines a “.ypkg” file type for YANG package definitions

Mounted packages

- Adds additional complexities:
 1. Removing mounted packages from included packages is tricky to model
 2. More/less specific mount paths (e.g., if some keys are specified)
 3. Much less industry experience with schema mount
- Latest solution:
 - Have a simple per mount flag (inherit-packages) to indicate whether to inherit mounted packages or start with a clean slate
 - Push more/less specific mount paths on to the package definer, i.e., treat as entirely separate paths.
 - Probably add list of supported-features under a mount-point

What's left?

- Expand & update examples:
 - Will require some tooling updates to validate these
- Finalize mount points:
 - Update description, add “supported-features”
- Define a package for instance-data schema
- A few more minor issues to resolve
- **Reviews from authors/WGs ...**

Next Steps

Meeting weekly on Tuesdays, 9 am US Eastern Time.

On the packages draft:

1. Review/tweak prose, finish examples
2. Complete remaining issues (TODOs in doc & github issues)

Related work:

- Make progress on tooling – hoping that I can open source

Any Questions/Comments?