Guidelines for YANG Example Validation and Coverage Analysis in IETF Documents

draft-cardona-claise-onion-yang-coverage-00

Camilo Cardona - NTT

Benoit Claise - Everything OPS

IETF 124, November 2025

Objective

- Propose tools and procedures to get the coverage of the modules of a path based on the examples provided in a draft.
- Coverage defined as: which leaves* from YANG models are included in the examples.

Why?

- Coverage provides information for authors and reviewers to identify missing parts in the examples of a document.
- Just like in programming, YANG model coverage is not the ultimate source of truth — but it helps.

* Not sure if leaves of leafs yet

Example, Schema:

This is a partial tree from a draft.

```
camilo@camilo-dev03:~/libyang/yang-coverage-tools/entitlements_example$ pyang -f tree  -p yang ietf-entitlement-inventory.yang
module: ietf-entitlement-inventory

  augment /inv:network-inventory/inv:network-elements/inv:network-element:
    +--ro installed-entitlements!
       +--ro entitlement* [entitlement-id]
          +--ro entitlement-id    -> /inv:network-inventory/ei:entitlements/entitlement/entitlement-id
          +--ro in-use?           boolean
  augment /inv:network-inventory/inv:network-elements/inv:network-element/inv:components/inv:component:
    +--ro installed-entitlements!
       +--ro entitlement* [entitlement-id]
          +--ro entitlement-id    -> /inv:network-inventory/ei:entitlements/entitlement/entitlement-id
          +--ro in-use?           boolean
  augment /inv:network-inventory/inv:network-elements/inv:network-element:
    +--ro capabilities!
       +--ro capability-class* [capability-class]
          +--ro capability-class    identityref
          +--ro capability* [capability-id]
             +--ro capability-id                     string
             +--ro extended-capability-description?   string
             +--ro entitlement-state!
             |  +--ro allowed?   boolean
             |  +--ro in-use?    boolean
             +--ro supporting-entitlement* [entitlement-id]
             |  +--ro entitlement-id    -> ../../../../../installed-entitlements/entitlement/entitlement-id
             +--ro capability-restriction* [capability-restriction-id]
                +--ro capability-restriction-id    string
                +--ro description?                 string
                +--ro resource-name?               string
                +--ro units?                       string
                +--ro max-value?                   int32
                +--ro current-value?               int32
  augment /inv:network-inventory/inv:network-elements/inv:network-element/inv:components/inv:component:
    +--ro capabilities!
       +--ro capability-class* [capability-class]
          +--ro capability-class    identityref
          +--ro capability* [capability-id]
             +--ro capability-id                     string
             +--ro extended-capability-description?   string
             +--ro entitlement-state!
             |  +--ro allowed?   boolean
             |  +--ro in-use?    boolean
             +--ro supporting-entitlement* [entitlement-id]
             |  +--ro entitlement-id    -> ../../../../../installed-entitlements/entitlement/entitlement-id
             +--ro capability-restriction* [capability-restriction-id]
                +--ro capability-restriction-id    string
                +--ro description?                 string
                +--ro resource-name?               string
                +--ro units?                       string
                +--ro max-value?                   int32
                +--ro current-value?               int32
  augment /inv:network-inventory:
    +--ro entitlements!
       +--ro entitlement* [entitlement-id]
          +--ro entitlement-id            string
          +--ro product-id?               string
          +--ro sku?                      string
          +--ro vendor?                   string
          +--ro part-number?              string
          +--ro state?                    entitlement-state-t
          +--ro renewal-profile
          |  +--ro activation-date?   yang:date-and-time
          |  +--ro start-date?        yang:date-and-time
          |  +--ro expiration-date?   yang:date-and-time
```

Example, Instances:

We have several examples for it.

```
camilo@camilo-dev03:~/libyang/yang-coverage-tools/entitlements_example$ ls examples
example1-basic-structure.json  example3-utilization-tracking.json       example5-license-pooling.json  example7-modular-components.json
example2-expired-license.json  example4-hierarchical-entitlements.json  example6-multi-vendor.json
```

Example, Tooling

```
camilo@camilo-dev03:~/libyang/yang-coverage-tools/entitlements_example$ ./entitlement_example2.sh
🧩 Extracting YANG coverage...
> yang-coverage-extract -p yang -p ... -o entitlements.coverage ... examples/*.json
Coverage saved to: entitlements.coverage
Covered paths: 53

📊 Analyzing coverage...
> yang-coverage-analyze -t -p yang -p ... -d ietf-entitlement-inventory.yang entitlements.coverage
🎯 Analyzing coverage for target module 'ietf-entitlement-inventory' and its augmentations only...
=== YANG Coverage Gap Analysis (Tree View) ===
Schema leaf nodes: 50
Visited nodes: 49
Coverage: 98.0%

Uncovered nodes in tree format:
module: ietf-entitlement-inventory (uncovered nodes only)
  +--rw network-inventory
     +--rw network-elements
        +--rw network-element
           +--rw components
              +--rw component
                 +--rw installed-entitlements
                    +--rw entitlement
                       +--rw in-use?
```

*https://yangson.labs.nic.cz/datamodel.html#yangson.datamodel.DataModel.ascii_tree

Example, YANGSON

Lada mentioned in the mailing list that Yangson* already supports a part of it

```
BrokenPipeError: [Errno 32] Broken pipe
Camilos-MacBook-Pro:yang camilo$ python print_tree.py | grep -B 3 --color=always "\{0\}"
grep: warning: stray \ before {
   |     +--ro vendor? {4}
   +--rw network-elements {7}
      +--rw network-element* [ne-id] {12}
         +--rw alias? {0}
--
         |     +--ro capability-class {12}
         +--rw components {12}
         |  +--rw component* [component-id] {15}
         |     +--rw alias? {0}
         |     +--ro asset-id? {0}
         |     +--ro breakout-channels! {0}
         |     |  +--ro breakout-channel* [channel-id] {0}
         |     |     +--ro channel-id {0}
--
         |     |     |  +--ro supporting-entitlement* [entitlement-id] {3}
         |     |     |     +--ro entitlement-id {3}
         |     |     +--ro capability-class {3}
         |     +--ro child-component-ref {0}
         |     +--ro class {15}
         |     +--rw component-id {15}
         |     +--rw description? {0}
         |     +--ro firmware-rev? {0}
         |     +--ro hardware-rev? {0}
         |     +--ro ietf-entitlement-inventory:installed-entitlements! {3}
         |     |  +--ro entitlement* [entitlement-id] {3}
         |     |     +--ro entitlement-id {3}
         |     |     +--ro in-use? {0}
         |     +--ro is-fru? {0}
         |     +--ro is-main? {0}
         |     +--ro mfg-date? {0}
         |     +--ro mfg-name? {0}
```


*https://yangson.labs.nic.cz/datamodel.html#yangson.datamodel.DataModel.ascii_tree

It is the path, not the destination

We need several things before we can automatically yield the coverage of a module:

- Automatic extraction of valid example files
- Differentiation from invalid examples
- Automatic linking/documentation of auxiliary YANG modules (ex: deviation, grouping)
- Validation of all valid examples
- Output of coverage results

Vision of How It Would work

For authors, it should be straightforward for *simple* cases:

- All models in the same document
- All models from other IETF documents

For *more complex* cases (deviations, schema-mount, yang-library), authors will need to do more work

- Extract auxiliary models
- Wrap examples in an instance data file (RFC 9195)

Gaps in Tooling

- The process of automatically validating an example based on document metadata and instance data files (RFC 9195) is probably the biggest gap
- Yangson already provides a base for coverage.
  - We might want to implement coverage in other tooling for diversity

Questions and Comments

- Interesting problem?
- To be solved in a generic way?

- My own questions: EXAMPLE BEGINS, RFC8407bis, datatracker, repo, …