Validate anydata with YANG Library context

draft-netana-nmop-yang-anydata-validation

ahmed.elhassany@swisscom.com
thomas.graf@swisscom.com

07. Nov. 2025

Problem statement

- Today, YANG data semantics under *anydata* subtree cannot be validated to assure data integrity.
- As example on the right, a network node publishing notifications in YANG-Push with a negative value for a counter type.
- Since anydata is used for publishing YANG data in notifications, subscribed YANG data can't be validated, and therefore YANG data processing chain can't be fully automated. This impacts YANG industry adoption.


```
{
 "ietf-yang-push:push-update": {
  "id": 89,
  "datastore-contents": {
   "ietf-interfaces:interfaces": {
   "interface": [
    {
     "name": "eth0",
              "statistics": {"in-octets": -10}
    }]
   }}
 }
}
```


Router is sending negative value for yang:counter64

Survey of existing use of "*anydata*"

- Operations on a YANG datastore
  1. A NETCONF subtree filter to select a subtree on an instantiated YANG data tree; RFC 8526, RFC 9144, and RFC 8641.
  2. The output of a subtree filter or XPATH; RFC 8526, RFC 9144, and RFC 8641.
  3. To represent edit operations on YANG data tree with YANG Patch; RFC 8072.
- Represent a YANG-like data tree (RFCs were not very clear)
  1. Subscribed notifications in RFC 8639.
  2. Error information in RESTCONF RFC 8040.

The goals of this document

- Enable automatic YANG data processing at scale.
- Provide an optional way to validate subscribed and published YANG data in notifications.
- YANG data consumers as in Section 4.7 of draft-ietf-nmop-yang-message-broker-integration can choose to enable or disable stricter validation in anydata subtrees.
- Data marked as invalid can be processed by a different data processing pipeline (e.g., logging for further analysis) and therefore not impacting the main business logic with misbehaving YANG data.

YANG Library look up

- The namespace of the encoded data nodes under *anydata* can be looked up in a YANG Library context.
- This approach **DOES NOT** restrict what a producer can put under *anydata* subtree.
- It only checks whatever subtree under the *anydata* have a corresponding schema and it is valid according to it.


```
{
"ietf-yang-library:yang-library": {
 "module-set": [
  {
   "name": "complete",
   "module": [
    {
     "name": "yang",
     "revision": "2022-06-16",
     "namespace": "urn:ietf:params:xml:ns:yang:1"
    },
    {
     "name": "ietf-interfaces",
     "revision": "2018-02-20",
     "namespace": "urn:ietf:params:xml:ns:yang:ietf-interfaces",
     "location": ["file://ietf-interfaces@2018-02-20.yang"],
     "feature": [
       "arbitrary-names",
       "pre-provisioning",
       "if-mib"
     ]
    },
   ...
```

Changes since IETF 123

1. Clarify the problem statement and the impact of having unvalidated *anydata* subtrees.
2. A survey of existing usage of *anydata* within IETF documents.

Context

- RFC 7950: The YANG 1.1 Data Modeling Language

  The "anydata" statement is used to represent an unknown set of nodes that can be modeled with YANG, except anyxml, but for which the data model is not known at module design time. It is possible for the data model, though not required, for anydata content to become known through protocol signaling or other means that are outside the scope of this document.