NETCONF
Internet-Draft
Updates: 8525 (if approved)
Intended status: Standards Track
Expires: 27 July 2026

Z. Lin
Huawei
B. Claise
Everything OPS
I. D. Martinez-Casanueva
Telefonica

23 January 2026

# Augmented-by Addition to the YANG Library
## draft-ietf-netconf-yang-library-augmentedby-15

## Abstract

"YANG Library" (RFC 8525) specifies the YANG module "ietf-yang-library" that provides information about the YANG modules, datastores, and datastore schemas used by a network management server.

This document augments the "ietf-yang-library" YANG data model to provide the augmented-by list. It facilitates the process of obtaining all dependencies between YANG modules, by querying ~~the~~ a network management server's YANG library. This document updates RFC 8525.

**Commenté [MB1]:** -- There is nothing changed in that spec

This is a conventional augmentation.

**Commenté [MB2]:** Which part is updated?

## Discussion Venues

This note is to be removed before publishing as an RFC.

Source for this draft and an issue tracker can be found at https://github.com/Zephyre777/draft-lincla-netconf-yang-library-augmentation.

## Status of This Memo



## Copyright Notice

Table of Contents

## 1. Introduction

The YANG library [RFC8525] provides information about the data models supported by ~~the~~ a server. This is presented as an inventory of YANG modules. It helps a client with listing all datastores supported by a network management server and the schema that is used by each of these datastores.

According to Sections 4.2.8 and 5.6.3 ~~in~~ of [RFC7950], augment defines additional nodes to ~~the~~ a module while deviations change (add, modify, delete) properties (sub-statements) to other module's schema nodes. They provide crucial information about the YANG data model composition, and is referred to as reverse dependency in this document: this means the behavior of a schema node depends on external modifications, creating flow backward from the augmenting/ deviation module to the base module.

**Commenté [MB3]:** How is this true for an augmented module?

These can be consumed Independent of external augmentations.

~~Currently, i~~It is difficult to obtain the YANG schema tree (defined in Section 3 in [RFC7950]) without obtaining and parsing all the YANG modules from a management server. The deviation list defined in YANG library enables ~~client~~ clients to obtain ~~the~~ a module reverse dependency without having to get and parse all YANG modules. However, the augmentation list is not defined in it.

Since both augmentation and deviation work as YANG module dependencies, it is reasonable to document them the same way in the YANG library. Having both augmentation and deviation directly available in the YANG library provides an easy and light~~-w~~weight solution for determining the reverse dependencies.

**Commenté [MB4]:** a convenient?

~~Therefore, t~~This document ~~proposes~~ specifies a YANG module that augments the YANG library to include the YANG module augmentation information.

One specific requirement with the implementation of this augmented-by

YANG module specification is, the modules and the augmented-by modules require ~~now~~ to be listed in the same module-set. Note that a similar requirement already applies for the YANG deviations.

### 1.1. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

### 1.2. Terminology

The terminology from [RFC8525] is used in this document

The term "client" is used as defined in [RFC6241] for NETCONF and [RFC8040] for RESTCONF.

The term "YANG schema tree" is used as defined ~~in the terminology in~~in Section X of [RFC7950]

Commenté [MB5]: Add section number

The term 'NMDA' (Network Management Datastore Architecture), which is defined as the title of [RFC8342] is used in this document.

Commenté [MB6]: ???

Tree diagrams in this document use the notations defined in [RFC8340].

This document defines the following term:

* Reverse Dependency: is the dependency YANG modules that provide external modification to the behavior of a base YANG module, by inserting additional nodes (augment) or deviating existing nodes (deviation).

## 2. Motivation

When using YANG modules, it is necessary to make sure that all its dependencies are present. [RFC7950] identifies four types of dependencies between YANG modules:

* Import: the "import" statement allows a module or submodule to reference definitions defined in other modules.

* Include: the "include" statement is used in a module to identify each submodule that belongs to it.

* Augment: the augmentation defines the location in the data model hierarchy where additional nodes are inserted.

* Deviation: the "deviation" statement defines a fragment of a module that the server does not implement.

~~The~~ "import" and "include" are direct dependencies which can be obtained by parsing ~~the~~ a YANG module source code, while the augmentation and deviation are reverse dependencies which are defined in another

module.

For the reverse dependencies, since they are defined externally, it is not possible to discover them by parsing ~~the~~ a YANG module. The current way to discover the reverse dependencies is to query all YANG modules from the server and parse them. This is a lengthy process, which must be repeated for each client that requires this information.

According to the definition of the module "ietf-yang-library" defined in [RFC8525], the "deviation" list describes that a module is deviated by which other modules. Since deviations and augmentations work similarly, if the YANG library could directly report all reverse dependencies, including augmentations, it would provide a much easier and light-weight solution to find module all dependencies, compared to obtaining and parsing all modules. In addition, to avoid causing problems in devices, for instance, falsely referring to modules in the wrong module-set, this document requires that the base modules and the augmentations be listed in the same module-set.

The YANG library only provides the deviation list, without augmentations. With augmentations being more widely used and defined, and with use cases to automate network management, augmentations become essential information for clients to better understand the network management server module relationships. Thus, the YANG library should be extended to also provide the augmentation information.

## 3. Use Cases

### 3.1. Data Mesh Architecture

As the demand for YANG-based telemetry [RFC8641] arises, there is a need for real-time knowledge of a specific YANG module's dependency list when a specific YANG-Push notification for a given subscription is received.

Some YANG-Push receivers will collect the information in advance of the telemetry collection, storing the entire module set for every single server who could be streaming data. However, this approach is not always practical in case of Configured Subscriptions [RFC8639] where the YANG-Push receiver is not configuring the subscriptions itself and in case of UDP transport [I-D.ietf-netconf-udp-notif]. See figure 1 in An Architecture for YANG-Push to Message Broker Integration [I-D.ietf-nmop-yang-message-broker-integration] for more details.

This architecture relies on the information of YANG dependencies in this specification to solve the problem of missing YANG semantics when notifications are transformed or indexed in Time Series Database.

Prior to the implementation of this specification, the method used for obtaining modules and finding module dependencies is to retrieve the full set of supported YANG modules from the network device, triggered by parsing the <subscription-started> message for each new YANG-Push subscription, because the YANG-push receiver would not know which YANG-Push publisher sends the subscribed YANG content until the

**Commenté [MB7]:** again

**Commenté [MB8]:** Thanks!

This is really useful to understand why/how

first notification is received.

By using the provided ~~the~~ augmentedby information in this specification, the YANG-Push receiver can directly obtain the YANG reverse dependencies for the specific YANG module(s) in the subscription by querying the server, saving collection and processing time at the YANG-Push receiver, by querying dependencies only for the required modules, and therefore helping with the real-time aspects of network observability.

Following is an example YANG-Push message of this use case received from within a subscription to YANG module "ietf-interfaces":

```
  "datastore-contents": {
    "ietf-interfaces:interfaces": [
    {
      "interface": {
        "name": "eth0",
        "type": "iana-if-type:ethernetCsmacd",
        "oper-status": "up",
        "speed": "1000000",
        "ietf-ip:ipv4": {
          "enabled": true,
          "forwarding": true
        }
      }
    }
    ]
  }
```

To correctly interpret the semantics of this message, both the "ietf-i~~I~~nterfaces" and "ietf-ip" YANG modules are required. Knowing only that the subscribed YANG module is "ietf-interfaces" is therefore insufficient, but that is currently the only dependency information exposed by the subscription information. In this case, a ~~lightweight~~ mechanism is needed to discover all relevant dependency modules, especially reverse dependencies (refer to "ietf-ip" in this example) so that every YANG module referenced in the pushed data can be reliably identified.

**Commenté [MB9]:** Cite RFCs as informative.

## 3.2. Data Catalog

Finding the YANG modules implemented by a network management server is paramount for configuring and monitoring the status of a network. However, since the inception of YANG the network industry has ~~experienced a tsunami~~defined large number of YANG modules developed by SDOs, open-source communities, and network vendors. This heterogeneity of YANG modules, that vary from one network device model to another, makes the management of a multi-vendor network a big challenge for operators~~.~~ [Martinez-Casanueva2023].

**Commenté [MB10]:** Why only device models are cited here?

In this regard, a data catalog provides a registry of the datasets exposed by remote data sources for consumers to discover data of interest. Besides the location of the dataset (i.e., the data

source), the data catalog registers additional metadata such as the data model (or schema) followed in the dataset or even related terms defined in a business glossary.

Data catalog solutions typically implement collectors that ingest metadata from the data sources themselves and external metadata sources. For example, a Kafka Schema Registry is a metadata source that provides metadata about the data model followed by some data stored in a Kafka topic.

In this sense, a YANG-enabled network device can be considered as another kind of data source, which the Data Catalog can pull metadata from. For instance, the data catalog can include a connector that fetches metadata about the YANG modules implemented by the network device. Combining this metadata with ~~other~~ others such as the business concept "interface", would enable data consumers to discover which datasets related to the concept "interface" are exposed by the network device.

Network devices that implement YANG library expose metadata about which YANG modules are implemented, and which are only imported. However, what a data consumer needs at the end are the YANG modules implemented by the device, hence, the combination of implemented YANG modules with other YANG modules that might deviate or augment the formers.

~~Coming back to the example of datasets related to the "interface" concept, say we have~~Consider a network device that implements the "ietf-interfaces" module [RFC8343] and the "ietf-ip" module [RFC8344], where the latter augments the former. For a data catalog to collect this metadata, a connector would retrieve YANG library data from the target device. However, the current version of YANG library would not satisfy the use case as it would tell that the device implements both "ietf-interfaces" and "ietf-ip" modules, but will miss the augment dependency between them.

~~The current~~A workaround is in combination with the YANG library data to additionally obtain both YANG modules and process them to discover that there is an augment dependency. This adds extra burden on the connector, which is forced to combine multiple metadata collection mechanisms. This process could be softened by extending YANG library to also capture augment dependencies, similarly to deviation dependencies.

## 4. The "ietf-yang-library-augmentedby" YANG module

This YANG module augments the "ietf-yang-library" module by adding the augmented-by leaf-list in the "yang-library/module-set" and "yang-library/modules-state". The name of list "augmented-by" indicates by which modules that the current module is being directly augmented.

The "yang-library/modules-state" is augmented despite its "deprecated" state to cope with the situation when container "modules-state" is used for compatibility reason with ietf-yang-library defined in [RFC7895]. Both the NMDA [RFC8342] and non-NMDA

compliant additions are defined in the same YANG module for simplicity for users and implementors.

For the scope of "augmented-by", this ~~draft~~ document only considers the direct augmentation relationship. The recursive result of augmentation or transitive dependency for module specified along the XPath, is out of the scope of this draft. Section 4.2 has given the implementation instructions.

**Commenté [MB11]:** I would put this text right after the tree diagram to understand the various augmented nodes.

### 4.1. Data Model Overview

#### 4.1.1. ~~YANG ietf-yang-library-augmentedby Tree View~~Tree Structure

The following is the YANG tree diagram for ~~model~~ module "ietf-yang-library-augmentedby".

```
module: ietf-yang-library-augmentedby

  augment /yanglib:yang-library/yanglib:module-set/yanglib:module:
    +-ro augmented-by*   -> ../../yanglib:module/name
  augment /yanglib:modules-state/yanglib:module:
    +-ro augmented-by*   -> ../../yanglib:module/name
```

#### 4.1.2. YANG ietf-yang-library-augmentedby Module

This YANG modules imports and augments the ~~modules~~ module "ietf-yang-library" [RFC8525].

**Commenté [MB12]:** One single module?

```
<CODE BEGINS> file "ietf-yang-library-augmentedby@2025-05-28.yang"
module ietf-yang-library-augmentedby {
  yang-version 1.1;
  namespace
    "urn:ietf:params:xml:ns:yang:ietf-yang-library-augmentedby";
  prefix yanglib-augby;

  import ietf-yang-library {
    prefix yanglib;
    reference
      "RFC 8525: YANG Library";
  }

  organization
    "IETF NETCONF (Network Configuration) Working Group";
  contact

    "WG Web:   <https://datatracker.ietf.org/wg/netconf/>
     WG List:  <mailto:netconf@ietf.org>

     Authors:   Zhuoyao Lin
                <mailto:zhuoyao.lin1@huawei-partners.com>
                Benoit Claise
                <mailto:benoit@everything-ops.net>
                Ignacio Dominguez Martinez-Casanueva
                <mailto:ignacio.dominguezmartinez@telefonica.com>";
  description
```

**Commenté [MB13]:** Meaninful prefix

```
        "This module augments the ietf-yang-library defined in
         {RFC8525} to provide not only the deviation list, but also
         the augmented-by list, in order to give sufficient
         information about the YANG modules reverse dependency. It
         facilitates the process of obtaining the entire
         dependencies of YANG module.

         The key words 'MUST', 'MUST NOT', 'REQUIRED', 'SHALL',
         'SHALL NOT', 'SHOULD', 'SHOULD NOT', 'RECOMMENDED',
         'NOT RECOMMENDED', 'MAY', and 'OPTIONAL' in this document
         are to be interpreted as described in BCP 14 (RFC 2119)
         (RFC 8174) when, and only when, they appear in all
         capitals, as shown here.

         Copyright (c) 2025-2026 IETF Trust and the persons identified as
         authors of the code.  All rights reserved.

         Redistribution and use in source and binary forms, with or
         without modification, is permitted pursuant to, and subject
         to the license terms contained in, the Revised BSD License
         set forth in Section 4.c of the IETF Trust's Legal Provisions
         Relating to IETF Documents
         (https://trustee.ietf.org/license-info).
         This version of this YANG module is part of RFC XXXX; see the
         RFC itself for full legal notices.  ";

      revision 2025-05-28 {
        description
          "Initial revision.";
        reference
          "RFC XXXX: Augmented-by Addition to the YANG Library";
      }

      grouping augmented-by {
        description
          "Augment the Provides augmented-by leaf-list from module info
with the
           module-augmented-by grouping";
        leaf-list augmented-by {
          type leafref {
            path "../../yanglib:module/yanglib:name";
          }
          description
            "Leaf-list of the augmentations used by this server to
             modify the schema tree of the module associated with
             this entry.  Note that the same module can be used for
             augmented-by for multiple modules, so the same
             entry MAY appear within multiple 'module' entries.

             This reference MUST NOT (directly or indirectly)
             refer to the module being augmented.

             Robust clients may want to make sure that they handle a
             situation where a module augments itself (directly or
             indirectly) gracefully.";
        }
      }
```

**Commenté [MB14]:** Avoid [XX] with the description

**Commenté [MB15]:** Follow template in RFC8407bis

**Commenté [MB16]:** Consider updating the description

**Commenté [MB17]:** --DISCUSS: What about the relationship with import-only-module?

Intuitively, this should not be present in that list?

```
    augment "/yanglib:yang-library/yanglib:module-set/yanglib:module" {
      description
        "Augments the augmented-by leaf-list from module info with the
         augmented-by grouping.";
      uses augmented-by;
    }

    augment "/yanglib:modules-state/yanglib:module" {
      status deprecated;
      description
        "Augments the augmented-by leaf-list from module info with the
         augmented-by grouping.";
      uses augmented-by;
    }
  }
  <CODE ENDS>
```

## 4.2. Implementation Instructions

### 4.2.1. The scope of augmented-by

This section explains the scope of augmented-by.

The "augmented-by" leaf-list should only consider those YANG modules that directly augment the YANG module in question in the ietf-yang-library, and the augmenting and augmented modules must be defined in the same module-set.

The "direct augment" is identified by the relationship between the augment module and the target node's parent module that it augments to. Only the direct parent module of the target node is augmented, and the rest of the parent modules defined in the schema tree are only indirect dependencies but not augmented modules. (Refer to "Target node" definition in Section 7.17 of [RFC7950])

In the case when a YANG application requires recursive dependency or specific schema tree dependency, the search logic should be implemented by the application itself.

A YANG example with the expected augmented-by result is provided in Section 4.2.2.

### 4.2.2. Examples

This section provides two module-set examples and their corresponding ietf-yang-library-augmentedby query results to explain the definition of "direct augment" stated in Section 4.2.1.

The two scenarios are provided with the same three YANG modules (Modules A, B, and C) defined in the module-set, however the relationships among them are different. All three YANG modules are defined in the same module-set name 'module-set1' to be able to augment or be augmented by the others. Otherwise, the yanglint validation should fail.

#### 4.2.2.1. Example 1

Relationships among Module A, Module B, and Module C in this example

**Commenté [MB18]:** Good point!

**Commenté [MB19]:** I would move this to be listed as a subsection o OPS CONS

**Commenté [MB20]:** I would move this to an appendix

**Commenté [MB21]:** Cite this is a specific implem example + cite a ref

are following:

*  Module A is the base module with container "foo-a"

*  Module B augments "/A:foo-a" with container "foo-b"

*  Module C augments "/A:foo-a" with leaf "leaf-c"

The ietf-yang-library-augmentedby query result is:

```
<CODE BEGINS> file "example_yanglib_result1.xml"
<yang-library xmlns="urn:ietf:params:xml:ns:yang:ietf-yang-library">
  <content-id>1</content-id>
  <module-set>
    <name>module-set1</name>
    <module>
      <name>A</name>
      <revision>2024-02-29</revision>
      <namespace>urn:ietf:params:xml:ns:yang:A</namespace>
      <augmented-by
      xmlns="urn:ietf:params:xml:ns:yang:
      ietf-yang-library-augmentedby">B</augmented-by>
      <augmented-by
      xmlns="urn:ietf:params:xml:ns:yang:
      ietf-yang-library-augmentedby">C</augmented-by>
    </module>
    <module>
      <name>B</name>
      <revision>2024-02-29</revision>
      <namespace>urn:ietf:params:xml:ns:yang:B</namespace>
    </module>
    <module>
      <name>C</name>
      <revision>2024-02-29</revision>
      <namespace>urn:ietf:params:xml:ns:yang:C</namespace>
    </module>
  </module-set>
</yang-library>
<modules-state xmlns="urn:ietf:params:xml:ns:yang:ietf-yang-library">
    <module-set-id>0</module-set-id>
</modules-state>
<CODE ENDS>
```

In this example, both Module B and Module C directly augment the container "foo-a". Therefore, both B and C are listed as "augmented-by" modules for Module A.

#### 4.2.2.2. Example 2

Relationships among Module A, Module B and Module C in this example are following:

*  Module A is the base module with container "foo-a"

*  Module B augments "/A:foo-a" with container "foo-b"

*  Module C augments "/A:foo-a/B:foo-b" with leaf "leaf-c"

The ietf-yang-library-augmentedby query result is:

```
<CODE BEGINS>
file "example_yanglib_result2.xml"
<yang-library xmlns="urn:ietf:params:xml:ns:yang:ietf-yang-library">
  <content-id>1</content-id>
  <module-set>
    <name>module-set1</name>
    <module>
      <name>A</name>
      <revision>2025-06-18</revision>
      <namespace>urn:ietf:params:xml:ns:yang:A</namespace>
      <augmented-by
      xmlns="urn:ietf:params:xml:ns:yang:
      ietf-yang-library-augmentedby">B</augmented-by>
    </module>
    <module>
      <name>B</name>
      <revision>2025-06-18</revision>
      <namespace>urn:ietf:params:xml:ns:yang:B</namespace>
      <augmented-by
      xmlns="urn:ietf:params:xml:ns:yang:
      ietf-yang-library-augmentedby">C</augmented-by>
    </module>
    <module>
      <name>C</name>
      <revision>2025-06-18</revision>
      <namespace>urn:ietf:params:xml:ns:yang:C</namespace>
    </module>
  </module-set>
</yang-library>
<modules-state xmlns="urn:ietf:params:xml:ns:yang:ietf-yang-library">
  <module-set-id>0</module-set-id>
</modules-state>
<CODE ENDS>
```

In this example, although the augment XPath statement used by Module C is rooted from the container "foo-a" defined in Module A, the node that Module C directly augments is the container "foo-b" defined in Module B. As a result, Module C is not considered to directly augment Module A and therefore does not appear in the "augmented-by" leaf-list of Module A. Only Module B, which directly augments the container "foo-a", is listed as an "augmented-by" module for Module A.

## 5. Operational Considerations

With the implementation of augmented-by, the base modules and the augmented-by modules require ~~now~~ to be listed in the same module-set.

**Commenté [MB22]:** Maintenance/update: The lib should be updated when new modules are onboarded. But this is similar to existing yang library.

Size: adding the reverse dependecny will increase the size of the library.

## 6. Implementation Status

Note to the RFC-Editor: Please remove this section before publishing.

### 6.1. Netopeer2 at IETF119 Hackathon

Zhuoyao Lin did the prototype implementation of the augmented-by list feature of this draft and demonstrated it based on Netopeer2 in IETF

119 Hackathon.

Netopeer2 is a NETCONF server & client implementation developed by CESNET. Source code is here: [NTP17]. The actual feature is implemented by extending the libyang [LY16] and sysrepo [SR16] which are the base libraries for Netopeer2 to support populating the augmented-by list.

### 6.2. Netopeer2 at IETF120 Hackathon

Zhuoyao Lin did a docker image of netopeer2 that integrates the augmented-by feauture in sysrepo and libyang. The result is presented at IETF 120 hackathon.

The source code can be obtained here: [NP24]

### 6.3. Libyangpush Find-dependency

Zhuoyao Lin did an implementation of find-dependency based on the ietf-yang-library with augmented-by feature in the YANG-Push message parser library libyangpush. The result is presented in IETF 120 hackathon.

The source code can be obtained here: [NP24]

### 6.4. Device Implementations

This section introduced the device implementations status [NA25] of this document from vendors like Huawei, Cisco, 6Wind.

The list of Router Images supporting (upcoming and already implemented) the ietf-yang-library-augmentedby YANG module proposed in this document is following:

* Huawei VRP R025C00 for NE8000 and NE40 (2025-10-15). https://support.huawei.com/

* Cisco IOS XR 25.3.1 (2025-08-31). https://software.cisco.com/

* Huawei VRP R024C10 for MA5800T-X17 (2025-06-30). https://support.huawei.com/

* 6WIND VSR 3.10. https://www.6wind.com/

## 7. Security Considerations

This section follows the guidelines in Section 3.7 of [I-D.ietf-netmod-rfc8407bis].

Since this document augments the ietf-yang-library YANG module defined in [RFC8525], the Section 6 Security Considerations of [RFC8525] apply here as well.

**Commenté [MB23]:** Redundant with the last para

The "ietf-yang-library-augmentedby" YANG module defines a data model that is designed to be accessed via YANG-based management protocols, such as NETCONF [RFC6241] and RESTCONF [RFC8040]. These YANG-based management protocols require the use of a secure transport layer (e.g., SSH [RFC4252], TLS [RFC8446], and QUIC [RFC9000]) and require

mutual authentication.

The Network Configuration Access Control Model (NACM) [RFC8341] provides a means to restrict access for particular NETCONF or RESTCONF users to a preconfigured subset of all available NETCONF or RESTCONF protocol operations and content.

There are no writable ("config true") data nodes defined in this YANG module.

This YANG module defines only readable ("config false") data nodes. Some of these readable data nodes may be considered sensitive or vulnerable in some network environments. It is important to control read access (e.g., via get, get-config, or notification) to these data nodes.

Specifically, the following readable data nodes contain potentially sensitive information:

* /yang-library/module-set/module/augmented-by

* /modules-state/module/augmented-by (modules-state is deprecated)

These nodes expose the augmentation relationships among modules implemented by a server. Unauthorized read access could help an attacker identify implementation structure, optional features, or software components present on a server, which might then be used to target known platform vulnerabilities. Therefore, access control policies SHOULD restrict read access to authorized users only.

There are no RPCs or action operations defined in this module. Therefore, there are no particularly sensitive RPC or action operations.

This module uses groupings from the ietf-yang-library YANG module defined in [RFC8525], which defines nodes that may be considered sensitive or vulnerable in network environments. Refer to the Section 6 Security Considerations of [RFC8525] for information as to which nodes may be considered sensitive or vulnerable in network environments.

## 8. IANA Considerations

This document registers one URI in the "IETF XML Registry" [RFC3688]. Following the format in [RFC3688], the following registration has been made.

URI: urn:ietf:params:xml:ns:yang:ietf-yang-library-augmentedby

Registration Contact: The IESG.

XML: N/A, the requested URI is an XML namespace.

This document registers one YANG module in the "YANG Module Names" registry [RFC6020]

Name: ietf-yang-library-augmentedby

Namespace: urn:ietf:params:xml:ns:yang:ietf-yang-library-augmentedby

Prefix: yanglib-aug

Reference: RFC-to-be (Note to the RFC-Editor: Please replace this with RFC plus the RFC number when this document can be published.)

Maintained by IANA: N

## 9. References

### 9.1. Normative References


[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC3688] Mealling, M., "The IETF XML Registry", BCP 81, RFC 3688, DOI 10.17487/RFC3688, January 2004, <https://www.rfc-editor.org/info/rfc3688>.

[RFC4252] Ylonen, T. and C. Lonvick, Ed., "The Secure Shell (SSH) Authentication Protocol", RFC 4252, DOI 10.17487/RFC4252, January 2006, <https://www.rfc-editor.org/info/rfc4252>.

[RFC6020] Bjorklund, M., Ed., "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", RFC 6020, DOI 10.17487/RFC6020, October 2010, <https://www.rfc-editor.org/info/rfc6020>.

[RFC6241] Enns, R., Ed., Bjorklund, M., Ed., Schoenwaelder, J., Ed., and A. Bierman, Ed., "Network Configuration Protocol (NETCONF)", RFC 6241, DOI 10.17487/RFC6241, June 2011, <https://www.rfc-editor.org/info/rfc6241>.

[RFC8040] Bierman, A., Bjorklund, M., and K. Watsen, "RESTCONF Protocol", RFC 8040, DOI 10.17487/RFC8040, January 2017, <https://www.rfc-editor.org/info/rfc8040>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

[RFC8340] Bjorklund, M. and L. Berger, Ed., "YANG Tree Diagrams", BCP 215, RFC 8340, DOI 10.17487/RFC8340, March 2018, <https://www.rfc-editor.org/info/rfc8340>.

[RFC8341] Bierman, A. and M. Bjorklund, "Network Configuration Access Control Model", STD 91, RFC 8341, DOI 10.17487/RFC8341, March 2018, <https://www.rfc-editor.org/info/rfc8341>.

[RFC8343] Bjorklund, M., "A YANG Data Model for Interface Management", RFC 8343, DOI 10.17487/RFC8343, March 2018, <https://www.rfc-editor.org/info/rfc8343>.

[RFC8344] Bjorklund, M., "A YANG Data Model for IP Management",


**Commenté [MB24]:** Info

**Commenté [MB25]:** Info

**Commenté [MB26]:** Infor

**Commenté [MB27]:** Infor

RFC 8344, DOI 10.17487/RFC8344, March 2018,
<https://www.rfc-editor.org/info/rfc8344>.

[RFC8446] Rescorla, E., "The Transport Layer Security (TLS) Protocol Version 1.3", RFC 8446, DOI 10.17487/RFC8446, August 2018, <https://www.rfc-editor.org/info/rfc8446>.

[RFC8525] Bierman, A., Bjorklund, M., Schoenwaelder, J., Watsen, K., and R. Wilton, "YANG Library", RFC 8525, DOI 10.17487/RFC8525, March 2019, <https://www.rfc-editor.org/info/rfc8525>.

[RFC9000] Iyengar, J., Ed. and M. Thomson, Ed., "QUIC: A UDP-Based Multiplexed and Secure Transport", RFC 9000, DOI 10.17487/RFC9000, May 2021, <https://www.rfc-editor.org/info/rfc9000>.

Commenté [MB28]: These were provided as examples

Commenté [MB29]: Info

Commenté [MB30]: Info

## 9.2. Informative References

[I-D.ietf-netconf-udp-notif]
Feng, A. H., Francois, P., Zhou, T., Graf, T., and P. Lucente, "UDP-based Transport for Configured Subscriptions", Work in Progress, Internet-Draft, draft-ietf-netconf-udp-notif-24, 20 November 2025, <https://datatracker.ietf.org/doc/html/draft-ietf-netconf-udp-notif-24>.

[I-D.ietf-netmod-rfc8407bis]
Bierman, A., Boucadair, M., and Q. Wu, "Guidelines for Authors and Reviewers of Documents Containing YANG Data Models", Work in Progress, Internet-Draft, draft-ietf-netmod-rfc8407bis-28, 5 June 2025, <https://datatracker.ietf.org/doc/html/draft-ietf-netmod-rfc8407bis-28>.

[I-D.ietf-nmop-yang-message-broker-integration]
Graf, T., "An Architecture for YANG-Push to Apache Kafka Integration", Work in Progress, Internet-Draft, draft-ietf-nmop-yang-message-broker-integration, 3 July 2024, <https://datatracker.ietf.org/doc/html/draft-ietf-nmop-yang-kafka-integration>.

[LY16] Vasko, M., "libyang", BSD-3-Clause license, November 2016, <https://github.com/CESNET/libyang.git>.

[Martinez-Casanueva2023]
Martinez-Casanueva, I. D., Gonzalez-Sanchez, D., Bellido, L., Fernandez, D., and D. R. Lopez, "Toward Building a Semantic Network Inventory for Model-Driven Telemetry", DOI 10.1109/MCOM.001.2200222, March 2023, <https://doi.org/10.1109/MCOM.001.2200222>. IEEE Communications Magazine

[NA25] "IETF YANG-Push - Implementations", <https://www.network-analytics.org/yp/implementations.html>.

[NP24] Lin, Z., "Netopeer2-docker-ietf120", July 2024, <https://github.com/network-

analytics/libyangpush/tree/feature/draft_augmentedby>.

[NTP17] Vasko, M., "Netopeer2", BSD-3-Clause license, May 2017, <https://github.com/CESNET/netopeer2>.

[RFC7895] Bierman, A., Bjorklund, M., and K. Watsen, "YANG Module Library", RFC 7895, DOI 10.17487/RFC7895, June 2016, <https://www.rfc-editor.org/info/rfc7895>.

[RFC7950] Bjorklund, M., Ed., "The YANG 1.1 Data Modeling Language", RFC 7950, DOI 10.17487/RFC7950, August 2016, <https://www.rfc-editor.org/info/rfc7950>.

[RFC8342] Bjorklund, M., Schoenwaelder, J., Shafer, P., Watsen, K., and R. Wilton, "Network Management Datastore Architecture (NMDA)", RFC 8342, DOI 10.17487/RFC8342, March 2018, <https://www.rfc-editor.org/info/rfc8342>.

[RFC8639] Voit, E., Clemm, A., Gonzalez Prieto, A., Nilsen-Nygaard, E., and A. Tripathy, "Subscription to YANG Notifications", RFC 8639, DOI 10.17487/RFC8639, September 2019, <https://www.rfc-editor.org/info/rfc8639>.

[RFC8641] Clemm, A. and E. Voit, "Subscription to YANG Notifications for Datastore Updates", RFC 8641, DOI 10.17487/RFC8641, September 2019, <https://www.rfc-editor.org/info/rfc8641>.

[SR16] Vasko, M., "sysrepo", BSD-3-Clause license, January 2016, <https://github.com/sysrepo/sysrepo.git>.

Commenté [MB31]: -- Normative

## Appendix A. Full Tree View of ietf-yang-library

The following is the YANG tree diagram [RFC8340] for module ietf-yang-library after adding augmentation from module ietf-yang-library-augmentedby. The RPCs and notifications are included as well.

```
module: ietf-yang-library
  +-ro yang-library
  |  +-ro module-set* [name]
  |  |  +-ro name                   string
  |  |  +-ro module* [name]
  |  |  |  +-ro name                        yang:yang-identifier
  |  |  |  +-ro revision?                   revision-identifier
  |  |  |  +-ro namespace                   inet:uri
  |  |  |  +-ro location*                   inet:uri
  |  |  |  +-ro submodule* [name]
  |  |  |  |  +-ro name         yang:yang-identifier
  |  |  |  |  +-ro revision?    revision-identifier
  |  |  |  |  +-ro location*    inet:uri
  |  |  |  +-ro feature*                    yang:yang-identifier
  |  |  |  +-ro deviation*                  -> ../../module/name
  |  |  |  +-ro yanglib-aug:augmented-by*
                                    -> ../../yanglib:module/name
  |  |  +-ro import-only-module* [name revision]
  |  |     +-ro name          yang:yang-identifier
  |  |     +-ro revision      union
  |  |     +-ro namespace     inet:uri
  |  |     +-ro location*     inet:uri
```

```
     |  |     +-ro submodule* [name]
     |  |        +-ro name        yang:yang-identifier
     |  |        +-ro revision?   revision-identifier
     |  |        +-ro location*   inet:uri
     |  +-ro schema* [name]
     |  |  +-ro name          string
     |  |  +-ro module-set*   -> ../../module-set/name
     |  +-ro datastore* [name]
     |  |  +-ro name      ds:datastore-ref
     |  |  +-ro schema    -> ../../schema/name


     |  +-ro content-id       string
     x--ro modules-state
        x--ro module-set-id    string
        x--ro module* [name revision]
           x--ro name                yang:yang-identifier
           x--ro revision            union
           +-ro schema?             inet:uri
           x--ro namespace           inet:uri
           x--ro feature*            yang:yang-identifier
           x--ro deviation* [name revision]
           |  x--ro name        yang:yang-identifier
           |  x--ro revision    union
           x--ro conformance-type    enumeration
           x--ro submodule* [name revision]
           |  x--ro name        yang:yang-identifier
           |  x--ro revision    union
           |  +-ro schema?     inet:uri
           +-ro yanglib-aug:augmented-by*   -> ../../yanglib:module/name

     notifications:
       +--n yang-library-update
       |  +-ro content-id    -> /yang-library/content-id
       x---n yang-library-change
          x--ro module-set-id    -> /modules-state/module-set-id
```


## Acknowledgments

The author would like to thank Jan Lindblad and Jean Quilbeuf for their help during the design of the YANG module, and Thomas Graf, Rob Wilton, Andy Bierman, Jean Quilbeuf, Alex Huang Feng, Per Andersson, Mahesh Jethanandani for their valuable review and comments.


## Authors' Addresses


Zhuoyao Lin
Huawei
Townsend Street, 4th Floor George's Court
Dublin
Ireland
Email: zhuoyao.lin1@huawei-partners.com

Benoit Claise
Everything OPS
Email: benoit@everything-ops.net

Ignacio Dominguez Martinez-Casanueva
Telefonica
Ronda de la Comunicacion, S/N
Madrid 28050
Spain
Email: ignacio.dominguezmartinez@telefonica.com