
Stream:	Internet Engineering Task Force (IETF)		
RFC:	10035		
Updates:	8525		
Category:	Standards Track		
Published:	August 2026		
ISSN:	2070-1721		
Authors:	Z. Lin	B. Claise	I. D. Martinez-Casanueva
	<i>Huawei</i>	<i>Everything OPS</i>	<i>Telefonica</i>

RFC 10035

YANG Library: Addition of the augmented-by List

Abstract

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

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

Status of This Memo

This is an Internet Standards Track document.

This document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and has been approved for publication by the Internet Engineering Steering Group (IESG). Further information on Internet Standards is available in Section 2 of RFC 7841.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at <https://www.rfc-editor.org/info/rfc10035>.

Copyright Notice

Copyright (c) 2026 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions

with respect to this document. Code Components extracted from this document must include Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

Table of Contents

1. Introduction	3
1.1. Requirements Language	3
1.2. Terminology	3
2. Motivation	4
3. Use Cases	5
3.1. Data Mesh Architecture	5
3.2. Data Catalog	6
4. The "ietf-yang-library-augmentedby" YANG Module	7
4.1. Data Model Overview	7
4.1.1. Tree Structure	7
4.1.2. The "ietf-yang-library-augmentedby" YANG Module	8
4.2. Implementation Instructions	10
4.2.1. The Scope of augmented-by	10
4.2.2. Examples	10
5. Operational Considerations	12
6. Security Considerations	13
7. IANA Considerations	13
8. References	14
8.1. Normative References	14
8.2. Informative References	14
Appendix A. Full Tree View of "ietf-yang-library"	16
Acknowledgments	17
Authors' Addresses	17

1. Introduction

The YANG library [RFC8525] provides information about the data models supported by a server. This is presented as an inventory of YANG modules. It helps a client by 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 of [RFC7950], "augment" defines additional nodes to a module, while a "deviation" changes (i.e., adds, modifies, or deletes) properties (sub-statements) of another module's schema nodes. They provide crucial information about the YANG data model composition, and this is referred to as "reverse dependency" in this document: this means the behavior of a schema node depends on external modifications, creating a backward flow from the augmenting and/or deviation module to the base module.

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

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 a convenient solution for determining the reverse dependencies.

This document specifies a YANG module that augments the YANG library to include the YANG module augmentation information. It updates [RFC8525] by modifying the list in Section 2 to also include the augmented-by list.

One specific requirement with the implementation of this augmented-by YANG module specification is that the modules and the augmented-by modules are required 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 Section 3 of [RFC7950].

The term "Network Management Datastore Architecture (NMDA)" is used as defined in [\[RFC8342\]](#).

Tree diagrams in this document use the notations defined in [\[RFC8340\]](#).

This document also defines the following term:

Reverse Dependency: the dependency on 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 of their 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 "augment" statement 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" statements are direct dependencies that can be obtained by parsing a YANG module's source code, while the "augment" and "deviation" statements are reverse dependencies that are defined in another module.

For the reverse dependencies, since they are defined externally, it is not possible to discover them by parsing the 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 "ietf-yang-library" module in [\[RFC8525\]](#), the "deviation" list identifies the name of each YANG module with deviation statements affecting the given YANG module. Since deviations and augmentations work similarly, if the YANG library could directly report all reverse dependencies, including augmentations, it would provide a much more convenient solution to find all module dependencies, as 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 rises for YANG-based telemetry [[RFC8641](#)], 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 that could be streaming data. However, this approach is not always practical in the case of configured subscriptions [[RFC8639](#)], where the YANG-Push receiver is not configuring the subscriptions itself, and in the case of UDP transport [[YANG-UDP](#)]. See Figure 1 in "An Architecture for YANG-Push to Message Broker Integration" [[YANG-PUSH](#)] 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 a 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, which is triggered by parsing the <subscription-started> message for each new YANG-Push subscription. This is because the YANG-Push receiver would not know which YANG-Push publisher sends the subscribed YANG content until the first notification is received.

By using the provided augmented-by 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. This saves collection and processing time at the YANG-Push receiver, by querying dependencies only for the required modules, therefore helping with the real-time aspects of network observability.

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

```
"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-interfaces" 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 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.

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 defined a large number of YANG modules developed by Standards Development Organizations (SDOs), open-source communities, and network vendors. This heterogeneity of YANG modules, which vary from one network device/service model to another, makes the management of a multi-vendor network a big challenge for operators [[Martinez-Casanueva2023](#)].

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 message broker schema registry is a metadata source that provides metadata about the data model followed by some data stored in a message broker topic.

In this sense, a YANG-enabled network device can be considered as another kind of data source, from which the data catalog can pull metadata. 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 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 the 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 former is needed.

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 the YANG library would not satisfy the use case, as it would tell that the device implements both "ietf-interfaces" and "ietf-ip" modules but would miss the augment dependency between them.

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

4. The "ietf-yang-library-augmentedby" YANG Module

4.1. Data Model Overview

4.1.1. Tree Structure

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

```
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:
    x--ro augmented-by*    -> ../../yanglib:module/name
```

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 the "augmented-by" list indicates by which modules the current module is being directly augmented.

The "yang-library/modules-state" is augmented despite its "deprecated" state to cope with the situation of when the "modules-state" container is used for compatibility reasons with the "ietf-yang-library" module 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 document only considers the direct augmentation relationship. The recursive result of augmentation or transitive dependency for modules specified along the XPath is out of the scope of this document. [Section 4.2](#) provides the implementation instructions.

4.1.2. The "ietf-yang-library-augmentedby" YANG Module

This YANG module imports and augments the "ietf-yang-library" module [[RFC8525](#)].

```
<CODE BEGINS> file "ietf-yang-library-augmentedby@2026-08-26.yang"

module ietf-yang-library-augmentedby {
  yang-version 1.1;
  namespace
    "urn:ietf:params:xml:ns:yang:ietf-yang-library-augmentedby";
  prefix yanglib-aug;

  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:zephyre888@gmail.com>
               Benoit Claise
               <mailto:benoit@everything-ops.net>
               Ignacio Dominguez Martinez-Casanueva
               <mailto:i.domingumar@gmail.com>";

  description
    "This module augments the 'ietf-yang-library' module defined in
    RFC 8525 to provide not only the deviation list, but also
    the augmented-by list, in order to give sufficient
    information about the YANG module's reverse dependency.  It
    facilitates the process of obtaining all the
    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) 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>).

All revisions of IETF and IANA published modules can be found
at the YANG Parameters registry group
(<https://www.iana.org/assignments/yang-parameters>).

This version of this YANG module is part of RFC 10035; see
the RFC itself for full legal notices.";

```
revision 2026-08-26 {
  description
    "Initial revision.";
  reference
    "RFC 10035: YANG Library: Addition of the augmented-by List";
}

augment "/yanglib:yang-library/yanglib:module-set/yanglib:module" {
  description
    "Adds augmented-by leaf-list to the list of modules of a
    module set";
  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 and MUST NOT
      be referenced in the import-only list.

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

augment "/yanglib:modules-state/yanglib:module" {
  status deprecated;
  description
    "Adds augmented-by leaf-list to the list of modules of a
    module set";
  leaf-list augmented-by {
    type leafref {
      path "../..//yanglib:module/yanglib:name";
    }
    status deprecated;
    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 and MUST NOT
    be referenced in the import-only list.

    Robust clients may want to make sure that they handle a
    situation where a module augments itself (directly or
    indirectly) gracefully.";
  }
}
}
<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" module, 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. 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; they are not augmented modules. (Refer to the definition of "target node" in [Section 7.17](#) of [\[RFC7950\]](#).)

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

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 (Module 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 YANG validation should fail.

The examples use line wrapping per [\[RFC8792\]](#).

4.2.2.1. Example 1

The relationships among Module A, Module B, and Module C in this example are as follows:

- 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"

<!-- NOTE: '\\' line wrapping per RFC 8792 -->
<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>2026-08-19</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>2026-08-19</revision>
      <namespace>urn:ietf:params:xml:ns:yang:B</namespace>
    </module>
    <module>
      <name>C</name>
      <revision>2026-08-19</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

The relationships among Module A, Module B, and Module C in this example are as follows:

- 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"

<!-- NOTE: '\\' line wrapping per RFC 8792 -->
<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>2026-08-19</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>2026-08-19</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>2026-08-19</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 are required to be listed in the same module-set.

Note that the reverse dependency will increase the size of the "ietf-yang-library" instance on the server. Also, the "ietf-yang-library" instance, and therefore the augmented-by instance, should be updated when new YANG modules are onboarded.

A YANG example with the expected augmented-by result is provided in [Section 4.2.2](#).

6. Security Considerations

This section is modeled after the template described in [Section 3.7.1](#) of [\[RFC9907\]](#).

The "ietf-yang-library-augmentedby" YANG module defines a data model that is designed to be accessed via YANG-based management protocols, such as the Network Configuration Protocol (NETCONF) [\[RFC6241\]](#) and RESTCONF [\[RFC8040\]](#). These YANG-based management protocols (1) have to use a secure transport layer (e.g., Secure Shell (SSH) [\[RFC4252\]](#), TLS [\[RFC9846\]](#), and QUIC [\[RFC9000\]](#)) and (2) have to use mutual authentication.

The Network Configuration Access Control Model (NACM) [\[RFC8341\]](#) provides the 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 the readable data nodes in this YANG module may be considered sensitive or vulnerable in some network environments. It is thus important to control read access (e.g., via get, get-config, or notification) to these data nodes. Specifically, the following subtrees and data nodes have particular sensitivities/vulnerabilities:

- /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. Since this document augments the "ietf-yang-library" YANG module defined in [\[RFC8525\]](#), the Security Considerations of [\[RFC8525\]](#) apply here as well. Refer to [Section 6](#) of [\[RFC8525\]](#) for information as to which nodes may be considered sensitive or vulnerable in network environments.

7. IANA Considerations

IANA has registered the following URI in the "ns" registry within the "IETF XML Registry" registry group [\[RFC3688\]](#):

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.

IANA has registered the following YANG module in the "YANG Module Names" registry [RFC6020] within the "YANG Parameters" registry group:

Name: ietf-yang-library-augmentedby

Maintained by IANA? N

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

Prefix: yanglib-aug

Reference: RFC 10035

8. References

8.1. 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>>.
- [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>>.
- [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>>.
- [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>>.
- [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>>.
- [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>>.

8.2. Informative References

[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", IEEE Communications Magazine, vol. 61, no. 3, pp. 60-66, DOI 10.1109/MCOM.001.2200222, March 2023, <<https://doi.org/10.1109/MCOM.001.2200222>>.
- [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>>.
- [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>>.
- [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>>.
- [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>>.
- [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>>.
- [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>>.
- [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", RFC 8344, DOI 10.17487/RFC8344, March 2018, <<https://www.rfc-editor.org/info/rfc8344>>.
- [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>>.
- [RFC8792] Watsen, K., Auerswald, E., Farrel, A., and Q. Wu, "Handling Long Lines in Content of Internet-Drafts and RFCs", RFC 8792, DOI 10.17487/RFC8792, June 2020, <<https://www.rfc-editor.org/info/rfc8792>>.
- [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>>.
- [RFC9846] Rescorla, E., "The Transport Layer Security (TLS) Protocol Version 1.3", RFC 9846, DOI 10.17487/RFC9846, July 2026, <<https://www.rfc-editor.org/info/rfc9846>>.

- [RFC9907]** Bierman, A., Boucadair, M., Ed., and Q. Wu, "Guidelines for Authors and Reviewers of Documents Containing YANG Data Models", BCP 216, RFC 9907, DOI 10.17487/RFC9907, March 2026, <<https://www.rfc-editor.org/info/rfc9907>>.
- [YANG-PUSH]** Graf, T. and A. Elhassany, "An Architecture for YANG-Push to Message Broker Integration", Work in Progress, Internet-Draft, draft-ietf-nmop-yang-message-broker-integration-13, 2 July 2026, <<https://datatracker.ietf.org/doc/html/draft-ietf-nmop-yang-message-broker-integration-13>>.
- [YANG-UDP]** 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-26, 29 July 2026, <<https://datatracker.ietf.org/doc/html/draft-ietf-netconf-udp-notif-26>>.

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

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

```
[NOTE: '\' line wrapping per RFC 8792]
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
    x--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 authors 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, and Med Boucadair for their valuable review and comments.

Authors' Addresses

Zhuoyao Lin

Huawei
Townsend Street, 4th Floor George's Court
Dublin
Ireland
Email: zephyre888@gmail.com

Benoit Claise

Everything OPS
Belgium
Email: benoit@everything-ops.net

Ignacio Dominguez Martinez-Casanueva

Telefonica

Ronda de la Comunicacion, S/N

28050 Madrid

Spain

Email: i.domingumar@gmail.com