

NETCONF
Internet-Draft
Updates: 8525 (if approved)
Intended status: Standards Track
Expires: 7 March 2026

Z. Lin
Huawei
B. Claise
Everything OPS
I. D. Martinez-Casanueva
Telefonica
3 September 2025

Augmented-by Addition to the YANG Library
draft-ietf-netconf-yang-library-augmentedby-12

Abstract

This document augments the ietf-yang-library 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.

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-linclanetconf-yang-library-augmentation>.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on 7 March 2026.

Copyright Notice

Copyright (c) 2025 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. Terminology	4
2. Motivation	5
3. Use Cases	6
3.1. Data Mesh Architecture	7
3.2. Data Catalog	8
4. The "ietf-yang-library-augmentedby" YANG module	9
4.1. Data Model Overview	10
4.1.1. Tree View	10
4.1.2. YANG Module	10
4.2. Implementation Instructions	13
4.2.1. The scope of augmented-by	13
4.2.2. An example of YANG module augmented-by result	13
5. Operational Considerations	14
6. Implementation Status	14
6.1. Netopeer2 at IETF119 Hackathon	15
6.2. Netopeer2 at IETF120 Hackathon	15
6.3. Libyangpush Find-dependency	15
6.4. Device Implementations	15
7. Changes	16
7.1. draft-lincla-netconf-yang-library-augmentation: Changes from 00 to 01	16
7.2. draft-lincla-netconf-yang-library-augmentedby version 00	16
7.3. draft-lincla-netconf-yang-library-augmentedby: Changes from 00 to 01	16
7.4. draft-lincla-netconf-yang-library-augmentedby: Changes from 01 to 02	16
7.5. draft-ietf-netconf-yang-library-augmentedby version 00	17
7.6. draft-ietf-netconf-yang-library-augmentedby: Changes from 00 to 01	17
7.7. draft-ietf-netconf-yang-library-augmentedby: Changes from 01 to 02	17
7.8. draft-ietf-netconf-yang-library-augmentedby: Changes from 02 to 03	17
7.9. draft-ietf-netconf-yang-library-augmentedby: Changes from 03 to 04	17

7.10. draft-ietf-netconf-yang-library-augmentedby: Changes from 04 to 05	18
7.11. draft-ietf-netconf-yang-library-augmentedby: Changes from 05 to 06	18
7.12. draft-ietf-netconf-yang-library-augmentedby: Changes from 06 to 07	18
7.13. draft-ietf-netconf-yang-library-augmentedby: Changes from 07 to 08	18
7.14. draft-ietf-netconf-yang-library-augmentedby: Changes from 08 to 09	19
7.15. draft-ietf-netconf-yang-library-augmentedby: Changes from 09 to 10	19
7.16. draft-ietf-netconf-yang-library-augmentedby: Changes from 10 to 11	19
8. Security Considerations	19
9. IANA Considerations	20
10. References	20
10.1. Normative References	20
10.2. Informative References	21
Appendix A. Full Tree View of ietf-yang-library	23
Appendix B. YANG module validation with yanglint	24
B.1. An ietf-yang-library yang-library data XML example	24
Contributors	26
Acknowledgments	26
Authors' Addresses	26

1. Introduction

The YANG library [RFC8525] provides information about the data model supported by the 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 Section 4.2.8 and 5.6.3 in [RFC7950], augment defines additional nodes to the 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.

Currently 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 to obtain the 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-weight solution for determining the reverse dependencies.

Therefore, this document proposes a YANG module that augments the YANG library to include the YANG module augmentation information.

One specific limitation 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 restriction already applies for the YANG deviations.

1.1. Terminology

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.

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 [RFC7950]

The term 'NMDA' which is defined as the title of [RFC8342] is used in this document.

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

This document defines the following term:

- * Reverse Dependency: The dependency YANG modules that provide modification to the behavior of a base YANG module's schema node, creating flow backward from the augmenting/deviation module to the base module, while they are not specified in the base module as opposed to the 'import' or 'include'.

2. Motivation

When using YANG modules, it is necessary to make sure that all its dependencies are presented. [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 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 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. If the YANG library could directly report all reverse dependencies, it would provide a much easier and light-weight solution to find module all dependencies, compared to obtaining and parsing all modules.

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.

From the perspective of operational efficiency, it is easy to adapt the device implementation to include augmentation, since augmentation and deviation work similarly. However, to avoid causing problems in devices, for instance, falsely referring to modules in the wrong module-set, the base modules and the augmentations require now to be listed in the same module-set.

3. Use Cases

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.

The alternative for a YANG-Push receiver is to collect and store the entire module set for every single server who could be streaming data. This approach is not always practical due to the following reasons:

- * A YANG-push receiver will not know which YANG-Push publisher sent the subscribed YANG content until the first notification is received.
- * For a YANG-Push receiver => from which YANG-Push publisher the subscribed YANG content is going to be received is not known until the first notification is being received.
- * Querying all the YANG modules is time-consuming and add overhead considering that only a subset of the YANG nodes of the management server are subscribed to.

This section introduces two use cases that reflect the motivation for extending the YANG library. One targets solving dependency problems in a data mesh architecture while the other aims at building a data catalog that makes YANG module information easily accessible.

3.1. Data Mesh Architecture

A network analytics architecture that integrates YANG-Push and Kafka is introduced in draft: An Architecture for YANG-Push to Apache Kafka Integration [I-D.ietf-nmop-yang-message-broker-integration]. This open-source project encompasses contributions such as Support of Versioning in YANG Notifications Subscription [I-D.ietf-netconf-yang-notifications-versioning] or Support of Network Observation Timestamping in YANG Notifications [I-D.netana-netconf-notif-envelope], among others.

The purpose of this project is to provide adequate information to the YANG-Push notifications so that the module and its dependencies can be parsed and retrieved automatically from the vantage point. The architecture relies on the information of YANG dependencies to realize, to solve the problem of missing YANG semantics when notifications are transformed or indexed in Time Series Database. As a solution to provide the missing YANG semantics, a schema registry is introduced to store YANG modules and all their relationships (Direct and reverse dependencies). The schema is obtained by NETCONF <get-schema> operation (defined in Section 3.1 of [RFC6022]) of the subscribed YANG schema tree, which is obtained by parsing the <subscription-started> message of each YANG-Push subscription.

When obtaining the dependency modules of a YANG module, an independent process containing multiple <get-schema> operations [RFC6022] is launched after each new YANG-Push subscription module has been known. However, the complexity remains at:

- * How dependencies of YANG modules are found (so that the YANG-Push subscription message has the complete set of module dependencies for its subscribed YANG schema tree)?
- * How do we conduct <get-schema>?

Currently, the method used for obtaining modules and finding module dependencies is getting the full set of supported YANG modules from the network device. This process is very heavy because in practice, each device may implement hundreds of YANG modules, requiring up to several minutes to complete the YANG modules retrieval. Besides, the need for parsing all YANG modules and finding all the dependencies adds an extra delay. Obtaining all dependencies through this method makes the operation very costly, since after each subscribed module is learned, the operation of getting the full set of modules needs to be performed again.

Therefore, considering the Network Observability real-time aspects, this extra delay in collecting (and processing) the dependencies through the approach of getting the full set of modules is not realistic.

It's more efficient to get dependencies only for the required modules.

By using the provided the augmentation information in ietf-yang-library, the YANG-Push receiver can directly obtain the YANG reverse dependencies by obtaining the contents of the YANG library, saving collection (and processing time) at the YANG-Push receiver and therefore helping with the real-time aspects of Network Observability and enabling closed-loop actions.

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 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]

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 models 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 these metadata with other 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 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 these 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 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, in a similar fashion 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/module-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 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.

4.1. Data Model Overview

4.1.1. Tree View

The following is the YANG tree diagram for model `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 Module

The YANG module source code of `ietf-yang-library-augmentedby` in which augmentation to the `ietf-yang-library` of [RFC8525] is defined.

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

  description
    "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 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 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 augmentation used by this server to
    modify the conformance 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.";
  }
}

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

    The 'augmented-by' leaf-list should only consider those YANG
    modules that directly augment the YANG module associated
    with this entry, and the augmenting and augmented modules
    must be defined in the same module-set.

    The 'directly 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)
    ";
  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.

    The 'augmented-by' leaf-list should only consider those YANG
    modules that directly augment the YANG module associated
    with this entry, and the augmenting and augmented modules
    must be defined in the same module-set.

    The 'directly 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)
    ";
  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. An example of YANG module augmented-by result

This section provides a module-set to explain the definition of "direct augment" stated in Section 4.2.1.

There are Modules A, B, C, which have the following relationships:

- * 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"

In this example, while the XPath that Module C augments to contains container "foo-a" defined in Module A, the actual target node it directly augments to is "foo-b" in Module B. Therefore, Module C is not an augmentation for Module A and does not appear in the "augmented-by" leaf-list of Module A. Only Module B, which is directly augmenting to "foo-a", is an "augmented-by" for Module A.

The augmented-by XML result for Modules A, B and C is the following:

```
<CODE BEGINS>
file "example_augmentedby_result.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-module</name>
      <revision>2025-06-18</revision>
      <namespace>urn:ietf:params:xml:ns:yang:a-module</namespace>
      <augmented-by
        xmlns="urn:ietf:params:xml:ns:yang:
        ietf-yang-library-augmentedby">b-module</augmented-by>
    </module>
    <module>
      <name>b-module</name>
      <revision>2025-06-18</revision>
      <namespace>urn:ietf:params:xml:ns:yang:b-module</namespace>
      <augmented-by
        xmlns="urn:ietf:params:xml:ns:yang:
        ietf-yang-library-augmentedby">c-module</augmented-by>
    </module>
    <module>
      <name>c-module</name>
      <revision>2025-06-18</revision>
      <namespace>urn:ietf:params:xml:ns:yang:c-module</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>
```

5. Operational Considerations

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

6. Implementation Status

Note to the RFC-Editor: Please remove this section before publishing (This follows the template in RFC7942).

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 feature 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. Changes

7.1. draft-lincla-netconf-yang-library-augmentation: Changes from 00 to 01

The list name has been updated from "augmentation" to "augmented-by", in order to represent the usage clearly.

The leafref has been changed from absolute path `/yanglib:yang-libraray/yanglib:module-set/yanglib:module/yanglib:name` to relative path `../../yanglib:module/yanglib:name`. The YANG validation in the appendix B shows that this path can work as expected.

Section 5 Implementation and section 6 Changes has been added.

7.2. draft-lincla-netconf-yang-library-augmentedby version 00

Updated the Use case content in Section 3.1. Add explanation: the scope of use case "Data Mesh Architecture" is limited to configured subscription.

Updated Implementation status content.

7.3. draft-lincla-netconf-yang-library-augmentedby: Changes from 00 to 01

Updated affiliations

Update content of Section 3.1 Data Mesh use case. Explain the limitation of applying get-all-schemas solution under the background of using UDP-notif of configured subscription, and how the feature proposed in the draft can improve the solution.

Full review of document. Nits and refinement of sections.

7.4. draft-lincla-netconf-yang-library-augmentedby: Changes from 01 to 02

Rewrite Section 2 Motivation.

Update Section 6 Changes's subsection title.

Update the Section 7 security consideration and section 8 IANA Considerations.

Added in the appendix the Impact Analysis of ietf-yang-library and proposal for the RFC8525bis draft.

7.5. draft-ietf-netconf-yang-library-augmentedby version 00

Resubmitted the draft name from:

draft-linclanet-netconf-yang-library-augmentedby-02

to:

draft-ietf-netconf-yang-library-augmentedby-00

7.6. draft-ietf-netconf-yang-library-augmentedby: Changes from 00 to 01

Correct the yanglint validation invalid example.

Updated the explanation to the yanglint validation example principle.

Delete Section "ietf-yang-library Impact Analysis, as an evaluation for RFC8525bis". The idea of updating the RFC8525 is paused.

7.7. draft-ietf-netconf-yang-library-augmentedby: Changes from 01 to 02

Update and rephrase the Introduction section.

Add Section 4.2 Implementation Instructions. Address in Section 4.2.1 that the definition of "augmented-by" only consider the direct augment. A YANG example for explaining this purpose has been put into Section 4.2.2.

Draft refinement.

Reference update.

7.8. draft-ietf-netconf-yang-library-augmentedby: Changes from 02 to 03

Merge review comment from Thomas.

7.9. draft-ietf-netconf-yang-library-augmentedby: Changes from 03 to 04

Update module content ietf-yang-library-augmentedby: Organise the augmentation content to grouping; Add augmentation to modules-state container.

Appendix B is deleted.

7.10. draft-ietf-netconf-yang-library-augmentedby: Changes from 04 to 05

Update ietf-yang-library-augmentedby module revision.

7.11. draft-ietf-netconf-yang-library-augmentedby: Changes from 05 to 06

Merged Jean, Thomas and Alex's editorial feedback.

Merged Andy's review.

Changed contents are listed following:

- * Section 1: Rewrite introduction. Added definition for 'reverse dependency'. Avoid possible misleading expression about reverse dependency and external dependency.
- * Section 2: Editorial Changes.
- * Section 3: Merged editorial feedback from Alex, mainly for Section 3.1.
- * Section 4: Move the full tree diagram to appendix A; Correct the YANG file date; Update the description statement in YANG module to provide implementation instructions information; Improve the text of implementation instruction and the direct augment example.
- * Appendix: For Appendix A: The ietf-yang-library full tree view has been moved to Appendix A. For Appendix B: The yanglint validation example has been moved from Appendix A to Appendix B. The invalid yanglint example has been removed. Set up texted has been added before the validation example.

7.12. draft-ietf-netconf-yang-library-augmentedby: Changes from 06 to 07

Change the reference in IANA Consideration to 'RFC-to-be'.

7.13. draft-ietf-netconf-yang-library-augmentedby: Changes from 07 to 08

Merged Per's review and comments.

7.14. draft-ietf-netconf-yang-library-augmentedby: Changes from 08 to 09

Updated YANG module's line length and spacing. Corrected the "author" to "authors".

Corrected typo.

Added section 5 Operational Considerations to explain that the base and augmentedby modules should be in the same module-set.

7.15. draft-ietf-netconf-yang-library-augmentedby: Changes from 09 to 10

Corrected the typo of 'requires'.

Moved RFC8525 reference to Normative References section.

7.16. draft-ietf-netconf-yang-library-augmentedby: Changes from 10 to 11

Mentioned in the title that this document updates RFC8525.

8. Security Considerations

The YANG module specified in this document defines a schema for data that is designed to be accessed via network management protocols such as NETCONF [RFC6241] or RESTCONF [RFC8040]. The lowest NETCONF layer is the secure transport layer, and the mandatory-to-implement secure transport is Secure Shell (SSH) [RFC6242]. The lowest RESTCONF layer is HTTPS, and the mandatory-to-implement secure transport is TLS [RFC8446].

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.

The readable node defined 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 this data node. The following is the explanation of data node's sensitivity/vulnerability:

The "augmented-by" list in this YANG module could reveal all modules that are augmenting one module. It could help attacker identify the relationship between modules and server implementations known bugs. Server vulnerabilities may include but not restricted to: 1. Too

many augmented-by records cause buffer overflow. 2. The augmented-by list helps identify through the inter-relation of modules how to cause the server to crash or significantly degrade device performance.

9. 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 NETCONF WG of the IETF.

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

10. References

10.1. 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>>.

- [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>>.
- [RFC6242] Wasserman, M., "Using the NETCONF Protocol over Secure Shell (SSH)", RFC 6242, DOI 10.17487/RFC6242, June 2011, <<https://www.rfc-editor.org/info/rfc6242>>.
- [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", 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>>.

10.2. Informative References

- [I-D.ietf-netconf-yang-notifications-versioning]
Graf, T., Claise, B., and A. H. Feng, "Support of
Versioning in YANG Notifications Subscription", Work in

Progress, Internet-Draft, draft-ietf-netconf-yang-notifications-versioning-08, 5 April 2025, <<https://datatracker.ietf.org/doc/html/draft-ietf-netconf-yang-notifications-versioning-08>>.

[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>>.

[I-D.netana-netconf-notif-envelope]

Feng, A. H., Francois, P., Graf, T., and B. Claise, "Extensible YANG Model for YANG-Push Notifications", Work in Progress, Internet-Draft, draft-netana-netconf-notif-envelope, <<https://datatracker.ietf.org/doc/html/draft-netana-netconf-notif-envelope>>.

[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>>.

[RFC6022] Scott, M. and M. Bjorklund, "YANG Module for NETCONF Monitoring", RFC 6022, DOI 10.17487/RFC6022, October 2010, <<https://www.rfc-editor.org/info/rfc6022>>.

[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>>.
- [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>>.

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

```

Appendix B. YANG module validation with yanglint

This section gives an example of the yang-library content that the user can try themselves with yanglint. This is created to demonstrate and validate the correct syntax. The example should be compiled with YANG modules ietf-yang-library and ietf-yang-library-augmentedby as schemas.

The examples provided are ietf-yang-library 'yang-library' data XML file containing the augmented-by field.

B.1. An ietf-yang-library yang-library data XML example

The yang-library XML example concerns the following module-set:

There are module1, module2, module3, which have the following relationships:

- * Module 1 is the base module with container "foo1"
- * Module 2 augments "/module1:foo1" with container "foo2"
- * Module 3 augments "/module1:foo1" with leaf "leaf3"

All three YANG modules are defined in the same module-set name 'ms1' to be able to augment or be augmented by the others. Otherwise, the yanglint validation should fail.

The example is following:

```
<CODE BEGINS> file "example_valid.xml"
<yang-library xmlns="urn:ietf:params:xml:ns:yang:ietf-yang-library">
  <content-id>1</content-id>
  <module-set>
    <name>ms1</name>
    <module>
      <name>module1</name>
      <revision>2024-02-29</revision>
      <namespace>urn:ietf:params:xml:ns:yang:module1</namespace>
      <augmented-by
        xmlns="urn:ietf:params:xml:ns:yang:
        ietf-yang-library-augmentedby">module2</augmented-by>
      <augmented-by
        xmlns="urn:ietf:params:xml:ns:yang:
        ietf-yang-library-augmentedby">module3</augmented-by>
    </module>
    <module>
      <name>module2</name>
      <revision>2024-02-29</revision>
      <namespace>urn:ietf:params:xml:ns:yang:module2</namespace>
    </module>
    <module>
      <name>module3</name>
      <revision>2024-02-29</revision>
      <namespace>urn:ietf:params:xml:ns:yang:module3</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>
```

Contributors

The following people all contributed to creating this document:

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 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