

Workgroup: NETMOD Working Group
Internet-Draft: draft-ietf-netmod-node-tags-08
Updates: [8407](#) (if approved)
Published: 22 June 2022
Intended Status: Standards Track
Expires: 24 December 2022
Authors: Q. Wu B. Claise P. Liu Z. Du
 Huawei Huawei China Mobile China Mobile
 M. Boucadair
 Orange

Node Tags in YANG Modules

Abstract

This document defines a method to tag nodes that are associated with operation and management data in YANG modules. This method for tagging YANG nodes is meant to be used for classifying either data nodes or instances of data nodes from different YANG modules and identifying their characteristic data. Tags may be registered as well as assigned during the definition of the module, assigned by implementations, or dynamically defined and set by users.

This document also provides guidance to future YANG data model writers; as such, this document updates RFC 8407.

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 24 December 2022.

Copyright Notice

Copyright (c) 2022 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
2.	Terminology
3.	Sample Use Cases for Node Tags
4.	Node Tag Values
4.1.	IETF Tags
4.2.	Vendor Tags
4.3.	User Tags
4.4.	Reserved Tags
5.	Node Tag Management
5.1.	Module Design Tagging
5.2.	Implementation Tagging
5.3.	User Tagging
6.	Node Tags Module Structure
6.1.	Node Tags Module Tree
7.	Node Tags YANG Module
8.	Guidelines to Model Writers
8.1.	Define Standard Tags
9.	IANA Considerations
9.1.	YANG Data Node Tag Prefixes Registry
9.2.	IETF YANG Data Node Tags Registry
9.3.	Updates to the IETF XML Registry
9.4.	Updates to the YANG Module Names Registry
10.	Security Considerations
11.	Acknowledgements
12.	Contributors
13.	References
13.1.	Normative References
13.2.	Informative References
Appendix A.	Example: Additional Auxiliary Data Property Information
Appendix B.	Instance Level Tunnel Tagging Example
Appendix C.	NETCONF Example
Appendix D.	Non-NMDA State Module
Appendix E.	Targeted Data Fetching Example
Appendix F.	Changes between Revisions
	Authors' Addresses

1. Introduction

The use of tags for classification and organization purposes is fairly ubiquitous, not only within IETF protocols, but globally in the Internet (e.g., "#hashtags"). For the specific case of YANG data models, a module tag is defined as a string that is associated with a module name at the module level [[RFC8819](#)].

Many data models have been specified by various Standards Developing Organizations (SDOs) and the Open Source community, and it is likely that many more will be specified. These models cover many of the networking protocols and techniques. However, data nodes defined by these technology-specific data models might represent only a portion of fault, configuration, accounting, performance, and security (FCAPS) management information ([\[FCAPS\]](#)) at different levels and network locations, but also categorized in various different ways. Furthermore, there is no consistent classification criteria or representations for a specific service, feature, or data source.

This document defines tags for both nodes in the schema tree and instance nodes in the data tree and shows how they can be associated with nodes within a YANG module, which:

- *Provide dictionary meaning for specific targeted data nodes;
- *Indicate a relationship between data nodes within the same YANG module or from different YANG modules;
- *Identify auxiliary data properties related to data nodes;
- *Identify key performance metric related data nodes and the absolute XPath expression identifying the element path to the nodes.

To that aim, this document defines a YANG module [[RFC7950](#)] that augments the YANG Module Tags ([\[RFC8819\]](#)) to provide a list of node entries to add or remove node tags as well as to view the set of node tags associated with specific data nodes or instance of data nodes within YANG modules. This new module is: "ietf-node-tags" (Section 7).

Typically, NETCONF clients can discover node tags supported by a NETCONF server by means of the <get-data> operation on the operational datastore (Section 3.1 of [[RFC8526](#)]) via the "ietf-node-tags" module. Alternatively, <get-schema> operation can be used to retrieve tags for nodes in the schema tree in any data module. These node tags can be used by a NETCONF [[RFC6241](#)] or RESTCONF [[RFC8040](#)] client to classify either data nodes or instance of these data nodes from different YANG modules and identify characteristic data and associated path to the nodes or node instances. Therefore, the

NETCONF/ RESTCONF client can query specific configuration or operational state on a server corresponding to characteristic data.

Similar to YANG module tags defined in [\[RFC8819\]](#), these node tags (e.g., tags for node in the schema node) may be registered or assigned during the module definition, assigned (e.g., tags for nodes in the data tree) by implementations, or dynamically defined and set by users. The contents of node tags from the operational state view are constructed using the following steps:

1. System tags (i.e., tags of "system" origin) that assigned during the module definition time are added;
2. User-configured tags (i.e., tags of "intended" origin) that dynamically defined and set by users at runtime;
3. Any tag that is equal to a masked-tag is removed.

This document defines an extension statement to indicate tags for data nodes. YANG metadata annotations are also defined in [\[RFC7952\]](#) as a YANG extension. The value of YANG metadata annotations is attached to a given data node instance and decided and assigned by the server and sent to the client (e.g., the origin value indicates to the client the origin of a particular data node instance) while tags for data node in the schema tree defined in Section 7 are retrieved centrally via the "ietf-node-tags" module and can be dynamically set by the client.

This document also defines an IANA registry for tag prefixes and a set of globally assigned tags ([Section 9](#)).

[Section 8](#) provides guidelines for authors of YANG data models. This document updates [\[RFC8407\]](#).

The YANG data model in this document conforms to the Network Management Datastore Architecture defined in [\[RFC8342\]](#).

2. 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 following terms are defined in [\[RFC7950\]](#) and are not redefined here:

*Data Node

*Data Tree

*Schema Tree

This document defines the following term:

Node Tag: Tag for YANG nodes used for classifying either data nodes or instances of data nodes from different YANG modules and identifying their characteristic data.

The meanings of the symbols in tree diagrams are defined in [[RFC8340](#)].

3. Sample Use Cases for Node Tags

The following lists a set of use cases to illustrate the use of node tags. This section does not intend to be exhaustive.

An example of the use of tags is to search discrete categories of YANG nodes that are scattered across the same or different YANG modules supported by a device. For example, if instances of these nodes in YANG modules are adequately tagged and set by a first client ("client A") via the "ietf-node-tags" module (Section 7) and retrieved by another client ("client B") from the operational datastore, then "client B" can obtain the path to the tagged nodes and subscribe only to network performance related data node instances in the operational datastore supported by a device.

"Client B" can also subscribe to updates from the operational datastore using the "ietf-node-tags" module. Any tag changes in the updates will then resynchronize to the "client B".

Also, tag classification is useful for users searching data nodes repositories. A query restricted to the "ietf:counter" data node tag in the "ietf-node-tags" module can be used to return only the YANG nodes that are associated with the counter. Without tags, a user would need to know the name of all the IETF YANG data nodes or instances of data nodes in different YANG modules.

Future management protocol extensions could allow for filtering queries of configuration or operational state on a server based on tags (for example, return all operational state related to system management).

4. Node Tag Values

All node tags (except in some cases of user tags as described in Section 4.3) begin with a prefix indicating who owns their definition. An IANA registry ([Section 9.1](#)) is used to register node tag prefixes. Initially, three prefixes are defined.

No further structure is imposed by this document on the value following the registered prefix, and the value can contain any YANG type 'string' characters except carriage returns, newlines, tabs, and spaces.

Except for the conflict-avoiding prefix, this document is purposefully not specifying any structure on (i.e., restricting) the tag values. The intent is to avoid arbitrarily restricting the values that designers, implementers, and users can use. As a result of this choice, designers, implementers, and users are free to add or not add any structure they may require to their own tag values.

4.1. IETF Tags

An IETF tag is a node tag that has the prefix "ietf:".

All IETF node tags are registered with IANA in the registry defined in [Section 9.2](#).

4.2. Vendor Tags

A vendor tag is a tag that has the prefix "vendor:".

These tags are defined by the vendor that implements the module, and are not registered with IANA. However, it is RECOMMENDED that the vendor includes extra identification in the tag to avoid collisions, such as using the enterprise or organization name following the "vendor:" prefix (e.g., vendor:entno:vendor-defined-classifier).

4.3. User Tags

User tags are defined by a user/administrator and are not registered by IANA.

Any tag with the prefix "user:" is a user tag. Furthermore, any tag that does not contain a colon (":", i.e., has no prefix) is also a user tag. Users are not required to use the "user:" prefix; however, doing so is RECOMMENDED.

4.4. Reserved Tags

Section 9.1 describes the IANA registry of tag prefixes. Any prefix not included in that registry is reserved for future use, but tags starting with such a prefix are still valid tags.

5. Node Tag Management

Tags may be associated with a data node within a YANG module in a number of ways. Typically, tags may be defined and associated at the module design time, at implementation time without the need of a

live server, or via user administrative control. As the main consumers of node tags are users, users may also remove any tag from a live server, no matter how the tag became associated with a data node within a YANG module.

5.1. Module Design Tagging

A data node definition MAY indicate a set of node tags to be added by a module's implementer. These design time tags are indicated using 'node-tag' extension statement.

If the data node is defined in an IETF Standards Track document, node tags MUST be IETF Tags ([Section 4.1](#)). Thus, new data nodes can drive the addition of new IETF tags to the IANA registry defined in [Section 9.2](#), and the IANA registry can serve as a check against duplication.

5.2. Implementation Tagging

An implementation MAY include additional tags associated with data nodes within a YANG module. These tags SHOULD be IETF ((i.e., registered)) or vendor tags.

5.3. User Tagging

Node tags of any kind, with or without a prefix, can be assigned and removed by the user from a server using normal configuration mechanisms. In order to remove a node tag from the operational datastore, the user adds a matching "masked-tag" entry for a given node within the 'ietf-node-tags' module.

6. Node Tags Module Structure

6.1. Node Tags Module Tree

The tree associated with the "ietf-node-tags" module is as follows:

```
module: ietf-node-tags
augment /tags:module-tags/tags:module:
  +-rw node-tags
    +-rw node* [id]
      +-rw id          nacm:node-instance-identifier
      +-rw tags* [tag]
        | +-rw tag      tags:tag
        | +-rw type?    identityref
      +-rw masked-tag* tags:tag
```

Figure 1: YANG Module Node Tags Tree Diagram

7. Node Tags YANG Module

The "ietf-node-tags" module imports types from [RFC8819] and [RFC8341].

<CODE BEGINS> file "ietf-node-tags@2022-02-04.yang"

```
module ietf-node-tags {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-node-tags";
  prefix ntags;

  import ietf-netconf-acm {
    prefix nacm;
    reference
      "RFC 8341: Network Configuration Access Control
        Model";
  }
  import ietf-module-tags {
    prefix tags;
    reference
      "RFC 8819: YANG Module Tags ";
  }

  organization
    "IETF NetMod Working Group (NetMod)";
  contact
    "WG Web: <https://datatracker.ietf.org/wg/netmod/>
    WG List: <mailto:netmod@ietf.org>

    Editor: Qin Wu
           <mailto:bill.wu@huawei.com>

    Editor: Benoit Claise
           <mailto:benoit.claise@huawei.com>

    Editor: Peng Liu
           <mailto:liupengyjy@chinamobile.com>

    Editor: Zongpeng Du
           <mailto:duzongpeng@chinamobile.com>

    Editor: Mohamed Boucadair
           <mailto:mohamed.boucadair@orange.com>;
  // RFC Ed.: replace XXXX with actual RFC number and
  // remove this note.
  description
    "This module describes a mechanism associating
    tags with YANG node within YANG modules. Tags may be IANA
    assigned or privately defined.

    Copyright (c) 2022 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 (<https://datatracker.ietf.org/html/rfcXXXX>); see the RFC itself for full legal notices.";

```
// RFC Ed.: update the date below with the date of RFC publication
// and RFC number and remove this note.
revision 2022-02-04 {
  description
    "Initial revision.";
  reference
    "RFC XXXX: Node Tags in YANG Modules";
}
identity node-tag-type {
  description
    "Base identity for node tag type.";
}
identity metric {
  base node-tag-type;
  description
    "Identity for metric tag type.";
}
identity delay {
  base node-tag-type;
  description
    "Identity for delay metric tag type.";
}
identity jitter {
  base node-tag-type;
  description
    "Identity for jitter metric tag type.";
}
identity loss {
  base node-tag-type;
  description
    "Identity for loss metric tag type.";
}
identity counter {
  base node-tag-type;
  description
    "Identity for counter metric tag type.";
}
identity summary {
  base node-tag-type;
```

```

    description
        "Identity for summary metric tag type.";
}
identity gauge {
    base node-tag-type;
    description
        "Identity for gauge metric tag type.";
}
identity unknown {
    base node-tag-type;
    description
        "Identity for unkown metric tag type.";
}
identity agg {
    base node-tag-type;
    description
        "Identity for aggregated metric tag type.";
}
extension node-tag {
    argument tag;
    description
        "The argument 'tag' is of type 'tag'. This extension statement
        is used by module authors to indicate node tags that should
        be added automatically by the system. As such, the origin of
        the value for the pre-defined tags should be set to 'system.'.";
}

augment "/tags:module-tags/tags:module" {
    description
        "Augment the Module Tags module with node tag
        attributes.";
    container node-tags {
        description
            "Contains the list of nodes or node instances and their associated
            node tags.";
        list node {
            key "id";
            description
                "Includes a list of nodes and their associated
                node tags.";
            leaf id {
                type nacm:node-instance-identifier;
                description
                    "The YANG data node name or data node instance name.";
            }
            list tags {
                key "tag";
                description
                    "Lists the tags associated with the node within

```

the YANG module.

See the IANA 'YANG node Tag Prefixes' registry for reserved prefixes and the IANA 'IETF YANG Data Node Tags' registry for IETF tags.

The 'operational' state view of this list is constructed using the following steps:

- 1) System tags (i.e., tags of 'system' origin) are added.
- 2) User configured tags (i.e., tags of 'intended' origin) are added.
- 3) Any tag that is equal to a masked-tag is removed."

reference

"RFC XXXX: node Tags in YANG Data
Modules, Section 9";

```
leaf tag {
  type tags:tag;
  description
    "Node tag corresponding to type of node tag.";
}
leaf type {
  type identityref {
    base node-tag-type;
  }
  description
    "Type of node tag.";
}
leaf-list masked-tag {
  type tags:tag;
  description
    "The list of tags that should not be associated with the
    node within the YANG module. The user can remove
    (mask) tags from the operational state datastore by
    adding them to this list. It is not an error to add tags
    to this list that are not associated with the data
    node within YANG module, but they have no operational
    effect.";
}
}
}
```

<CODE ENDS>

8. Guidelines to Model Writers

This section updates [[RFC8407](#)] by providing text that may be regarded as a new subsection to Section 4 of that document. It does not change anything already present in [[RFC8407](#)].

8.1. Define Standard Tags

A module MAY indicate, using node tag extension statements, a set of node tags that are to be automatically associated with node within the module (i.e., not added through configuration).

```
module example-module-A {  
  //...  
  import ietf-node-tags { prefix ntags; }  
  
  container top {  
    list X {  
      leaf foo {  
        ntags:node-tag "ietf:summary";  
      }  
      leaf bar {  
        ntags:node-tag "ietf:loss";  
      }  
    }  
  }  
  // ...  
}
```

Figure 2: An Example of Data Object Tag

The module writer can use existing standard node tags, or use new node tags defined in the data node definition, as appropriate. For IETF standardized modules, new node tags MUST be assigned in the IANA registry defined in [Section 9.2](#).

9. IANA Considerations

9.1. YANG Data Node Tag Prefixes Registry

This document requests IANA to create "YANG node Tag Prefixes" subregistry in "YANG node Tag" registry.

Prefix entries in this registry should be short strings consisting of lowercase ASCII alpha-numeric characters and a final ":" character.

The allocation policy for this registry is Specification Required [[RFC8126](#)]. The Reference and Assignee values should be sufficient to

identify and contact the organization that has been allocated the prefix. There is no specific guidance for the Designated Expert and there is a presumption that a code point should be granted unless there is a compelling reason to the contrary.

The initial values for this registry are as follows:

Prefix	Description	Reference	Assignee
ietf:	IETF Tags allocated in the IANA IETF YANG node Tags registry	[This document]	IETF
vendor:	Non-registered tags allocated by the module's implementer.	[This document]	IETF
user:	Non-registered tags allocated by and for the user.	[This document]	IETF

Figure 3: Table 1

Other standards organizations (SDOs) wishing to allocate their own set of tags should request the allocation of a prefix from this registry.

9.2. IETF YANG Data Node Tags Registry

This document requests IANA to create "IETF Node Tags" subregistry in "YANG node Tag" registry. This subregistry appears below "YANG node Tag Prefixes" registry.

This subregistry allocates tags that have the registered prefix "ietf:". New values should be well considered and not achievable through a combination of already existing IETF tags.

The allocation policy for this subregistry is IETF Review [[RFC8126](#)]. The Designated Expert is expected to verify that IANA assigned tags conform to Net-Unicode as defined in [[RFC5198](#)], and shall not need normalization.

The initial values for this subregistry are as follows:

Node Tag	Description	Reference
ietf:metric	Represent metric data (e.g., ifstatistics) associated with specific node (e.g., interfaces)	[This document]
ietf:delay	Represents the delay metric data associated with specific node.	[This document]
ietf:jitter	Represents the jitter metric data associated with specific node.	[This document]
ietf:loss	Represents the loss metric data associated with specific node.	[This document]
ietf:counter	Represents any metric value associated with specific node that monotonically increases over time, starting from zero.	[This document]
ietf:gauge	Represents current measurements associated with specific node that may increase, decrease or stay constant.	[This document]
ietf:summary	Represents the metric value associated with specific node that measures distributions of discrete events without knowing predefined range.	[This document]
ietf:unknown	Represents the metric value associated with specific node that can not determine the type of metric.	[This document]
ietf:agg	Relates to aggregated metric value associated with specific node (i.e.,	[This document]

| aggregated statistics) |
+-----+-----+-----+

Figure 4: Table 2

A data node can contain one or multiple node tags. Data node to be tagged with the initial value in Table 2 can be one of 'container', 'leaf-list', 'list', or 'leaf' data node. All tag values described in Table 2 can be inherited down the containment hierarchy if Data nodes tagged with those tag values is one of 'container', 'leaf-list', 'list'.

9.3. Updates to the IETF XML Registry

This document registers the following namespace URI in the "ns" subregistry within the "IETF XML Registry" [[RFC3688](#)]:

URI: urn:ietf:params:xml:ns:yang:ietf-node-tags
Registrant Contact: The IESG.
XML: N/A; the requested URI is an XML namespace.

9.4. Updates to the YANG Module Names Registry

This document registers the following YANG module in the YANG Module Names registry [[RFC6020](#)] within the "YANG Parameters" registry:

name: ietf-node-tags
namespace: urn:ietf:params:xml:ns:yang:ietf-node-tags
prefix: ntags
reference: RFC XXXX
maintained by IANA: N

10. Security Considerations

The YANG module specified in this document defines 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, e.g., the presence of tags may reveal information about the way in which data nodes or node instances are used and therefore providing access to private information or revealing an attack vector should be restricted. Note that appropriate privilege and security levels need to be applied to the addition and removal of user tags to ensure that a user receives the correct data.

This document adds the ability to associate node tag with data nodes or instances of data nodes within the YANG modules. This document does not define any actions based on these associations, and none are yet defined, and therefore it does not by itself introduce any new security considerations.

Users of the node tag meta-data may define various actions to be taken based on the node tag meta-data. These actions and their definitions are outside the scope of this document. Users will need to consider the security implications of any actions they choose to define, including the potential for a tag to get 'masked' by another user.

11. Acknowledgements

The authors would like to thank Ran Tao for his major contributions to the initial modeling and use cases.

The authors would also like to acknowledge the comments and suggestions received from Juergen Schoenwaelder, Andy Bierman, Lou Berger, Jaehoon Paul Jeong, Wei Wang, Yuan Zhang, Ander Liu, YingZhen Qu, Boyuan Yan, Adrian Farrel, and Mahesh Jethanandani.

12. Contributors

Liang Geng
Individual
32 Xuanwumen West St, Xicheng District
Beijing 10053

13. References

13.1. 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>>.
- [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>>.
- [RFC8126] Cotton, M., Leiba, B., and T. Narten, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 8126, DOI 10.17487/RFC8126, June 2017, <<https://www.rfc-editor.org/info/rfc8126>>.
- [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>>.
- [RFC8407] Bierman, A., "Guidelines for Authors and Reviewers of Documents Containing YANG Data Models", BCP 216, RFC 8407, DOI 10.17487/RFC8407, October 2018, <<https://www.rfc-editor.org/info/rfc8407>>.
- [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>>.
- [RFC8819] Hopps, C., Berger, L., and D. Bogdanovic, "YANG Module Tags", RFC 8819, DOI 10.17487/RFC8819, January 2021, <<https://www.rfc-editor.org/info/rfc8819>>.

13.2. Informative References

- [FCAPS] International Telecommunication Union, "X.700 : Management framework for Open Systems Interconnection

(OSI) for CCITT applications", , September 1992, <<http://www.itu.int/rec/T-REC-X.700-199209-I/en>>.

[RFC5198] Klensin, J. and M. Padlipsky, "Unicode Format for Network Interchange", RFC 5198, DOI 10.17487/RFC5198, March 2008, <<https://www.rfc-editor.org/info/rfc5198>>.

[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>>.

[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>>.

[RFC7952] Lhotka, L., "Defining and Using Metadata with YANG", RFC 7952, DOI 10.17487/RFC7952, August 2016, <<https://www.rfc-editor.org/info/rfc7952>>.

[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>>.

[RFC8526] Bjorklund, M., Schoenwaelder, J., Shafer, P., Watsen, K., and R. Wilton, "NETCONF Extensions to Support the Network Management Datastore Architecture", RFC 8526, DOI 10.17487/RFC8526, March 2019, <<https://www.rfc-editor.org/info/rfc8526>>.

[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>>.

[RFC9195] Lengyel, B. and B. Claise, "A File Format for YANG Instance Data", RFC 9195, DOI 10.17487/RFC9195, February 2022, <<https://www.rfc-editor.org/info/rfc9195>>.

[RFC9196] Lengyel, B., Clemm, A., and B. Claise, "YANG Modules Describing Capabilities for Systems and Datastore Update Notifications", RFC 9196, DOI 10.17487/RFC9196, February 2022, <<https://www.rfc-editor.org/info/rfc9196>>.

Appendix A. Example: Additional Auxiliary Data Property Information

This section gives an example of how Auxiliary Data Property Module could be defined. It demonstrates how auxiliary data property configuration parameters can be conditionally augmented to the generic node list. The example is not intended as a complete module for Auxiliary Data Property configuration.

```

module ex-auxiliary-data-property {
  yang-version 1.1;
  namespace "http://example.com/auxiliary-data-property";
  prefix "dp";

  import ietf-module-tags {
    prefix tags;
  }
  import ietf-node-tags {
    prefix ntags;
  }
  identity critical {
    base ntags:node-tag-type;
    description
      "Identity for critical node tag type.";
  }
  augment "/tags:module-tags/tags:module/ntags:node-tags/ntags:"
    + "node/ntags:tags" {
    when 'derived-from-or-self(ntags:type, "dp:critical")';
    description "Extend ietf-node-tags module for auxiliary data prop
leaf value {
      type string;
      description
        "The auxiliary information corresponding
        to data node instance tagged with 'critical'
        node tag type.";
    }
    // other auxiliary data property config params, etc.
  }
}

```

Appendix B. Instance Level Tunnel Tagging Example

In the example shown in the following figure, the 'tunnel-svc' data node is a list node defined in a 'example-tunnel-pm' module and has 7 child nodes: 'name', 'create-time', 'modified-time', 'average-latency', 'packet-loss', 'min-latency', 'max-latency' leaf node. In these child nodes, the 'name' leaf node is the key leaf for the 'tunnel-svc' list. Following is the tree diagram [RFC8340] for the "example-tunnel-pm" module:

```

+--rw tunnel-svc* [name]
|   +--rw name                string
|   +--ro create-time         yang:date-and-time
|   +--ro modified-time       yang:date-and-time
|   +--ro average-latency     yang:gauge64
|   +--ro packet-loss         yang:counter64
|   +--ro min-latency         yang:gauge64
|   +--ro max-latency         yang:gauge64

```

To help identify specific data for a customer, users tags on specific instances of the data nodes are created as follows:

```
<rpc message-id="103"
  xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <edit-data xmlns="urn:ietf:params:xml:ns:yang:ietf-netconf-nmda"
    xmlns:ds="urn:ietf:params:xml:ns:yang:ietf-datastores">
    <datastore>ds:running</datastore>
    <config>
    <module-tag>
    <module>
    <name>example-tunnel-pm</name>
    <node-tags
      xmlns="urn:ietf:params:xml:ns:yang:ietf-node-tags">
    <node>
    <id>
      /tp:tunnel-svc[name='foo']/tp:packet-loss
    </id>
    <tags>
    <tag>user:customer1_example_com</tag>
    </tags>
    <tags>
    <tag>ietf:critical</tag>
    </tags>
    </node>
    <node>
    <id>
      /tp:tunnel-svc[name='bar']/tp:modified-time
    </id>
    <tags>
    <tag>user:customer2_example_com</tag>
    </tags>
    </node>
    </node-tags>
    </module>
    </module-tag>
    </config>
  </edit-data>
</rpc>
```

Note that the 'ietf:critical' tag is additional new tag value that needs to be allocated from "IETF Node Tags" subregistry in Section 9.2.

Appendix C. NETCONF Example

The following is a NETCONF example result from a query of node tags list. For the sake of brevity only a few module and associated data

node results are provided. The example uses the folding defined in [\[RFC8792\]](#).

===== NOTE: '\' line wrapping per RFC 8792 =====

```
<ns0:data xmlns:ns0="urn:ietf:params:xml:ns:netconf:base:1.0">
  <t:module-tags xmlns:t="urn:ietf:params:xml:ns:yang:ietf-module-tags">
    <t:module>
      <t:name>ietf-interfaces</t:name>
      <s:node-tags
        xmlns:s="urn:ietf:params:xml:ns:yang:ietf-node-tags">
        <s:node>
          <s:id>
            /if:interfaces/if:interface/if:statistics/if:in-errors
          </s:id>
          <s:tags>
            <s:tag>ietf:metric</s:tag>
          </s:tags>
          <s:tags>
            <s:tag>ietf:loss</s:tag>
          </s:tags>
          <s:tags>
            <s:tag>ietf:agg</s:tag>
          </s:tags>
        </s:node>
      </s:node-tags>
    </t:module>
    <t:module>
      <t:name>ietf-ip</t:name>
      <s:node-tags
        xmlns:s="urn:ietf:params:xml:ns:yang:ietf-node-tags">
        <s:node>
          <s:id>/if:interfaces/if:interface/ip:ipv4/ip:mtu</s:id>
          <s:tags>
            <s:tag>ietf:metric</s:tag>
          </s:tags>
        </s:node>
      </s:node-tags>
    </t:module>
  </t:module-tags>
</ns0:data>
```

Figure 5: Example NETCONF Query Output

Appendix D. Non-NMDA State Module

As per [RFC8407], the following is a non-NMDA module to support viewing the operational state for non-NMDA compliant servers.

<CODE BEGINS> file "ietf-node-tags-state@2022-02-03.yang"

```
module ietf-node-tags-state {
  yang-version 1.1;
  namespace
    "urn:ietf:params:xml:ns:yang:ietf-node-tags-state";
  prefix ntags-s;

  import ietf-netconf-acm {
    prefix nacm;
    reference
      "RFC 8341: Network Configuration Access Control
        Model";
  }
  import ietf-module-tags {
    prefix tags;
  }
  import ietf-module-tags-state {
    prefix tags-s;
    reference
      "RFC 8819: YANG Module Tags ";
  }
  organization
    "IETF NetMod Working Group (NetMod)";

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

    Editor: Qin Wu
           <mailto:bill.wu@huawei.com>

    Editor: Benoit Claise
           <mailto:benoit.claise@huawei.com>

    Editor: Peng Liu
           <mailto:liupengyjy@chinamobile.com>

    Editor: Zongpeng Du
           <mailto:duzongpeng@chinamobile.com>

    Editor: Mohamed Boucadair
           <mailto:mohamed.boucadair@orange.com>";
  // RFC Ed.: replace XXXX with actual RFC number and
  // remove this note.
  description
    "This module describes a mechanism associating data node
      tags with YANG data node within YANG modules. Tags may be
      IANA assigned or privately defined."
```

Copyright (c) 2022 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 Simplified 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 (<https://datatracker.ietf.org/html/rfcXXXX>); see the RFC itself for full legal notices.";

```
// RFC Ed.: update the date below with the date of RFC publication
// and RFC number and remove this note.
revision 2022-02-04 {
  description
    "Initial revision.";
  reference
    "RFC XXXX: Node Tags in YANG Data
      Modules";
}
identity node-tag-type {
  description
    "Base identity for node tag type.";
}
augment "/tags-s:module-tags-state/tags-s:module" {
  description
    "Augments the Module Tags module with node tag
      attributes.";
  container node-tags {
    config false;
    status deprecated;
    description
      "Contains the list of data nodes and their
        associated self describing tags.";
    list node {
      key "id";
      status deprecated;
      description
        "Lists the data nodes and their associated self
          describing tags.";
      leaf id {
        type nacm:node-instance-identifier;
        mandatory true;
        status deprecated;
        description
          "The YANG data node name.";
```

```

    }
list tags {
    key "tag";
    status deprecated;
    description
        "Lists the tags associated with the data node within
        the YANG module.

        See the IANA 'YANG node Tag Prefixes' registry
        for reserved prefixes and the IANA 'IETF YANG Data
        Node Tags' registry for IETF tags.

        The 'operational' state view of this list is
        constructed using the following steps:

        1) System tags (i.e., tags of 'system' origin) are
            added.
        2) User configured tags (i.e., tags of 'intended'
            origin) are added.
        3) Any tag that is equal to a masked-tag is removed.";
    reference
        "RFC XXXX: Node Tags in YANG Data
        Modules, Section 9";
leaf tag {
    type tags:tag;
    status deprecated;
    description
        "Node tag corresponding to type of node tag.";
}
leaf type {
    type identityref {
        base node-tag-type;
    }
    status deprecated;
    description "type of the node tag.";
}
}
leaf-list masked-tag {
    type tags:tag;
    status deprecated;
    description
        "The list of tags that should not be associated with the
        data node within the YANG module. The user can remove
        (mask) tags from the operational state datastore by
        adding them to this list. It is not an error to add
        tags to this list that are not associated with the
        data node within YANG module, but they have no
        operational effect.";
}

```

```

    }
  }
}

```

<CODE ENDS>

Appendix E. Targeted Data Fetching Example

The following provides tagged data node Fetching example. The subscription "id" values of 22 used below is just an example. In production, the actual values of "id" might not be small integers.

```

+-----+
| Subscriber|
+-----+
|
|   Node Tagging Fetching
|   (id, node-tag = metric)
|<-----+
|
|   establish-subscription
+----->|
|
|   RPC Reply: OK, id = 22
|<-----+
|
|   Notification Message (for 22)
|<-----+
|
|

```

The subscriber can query node tag list from operational datastore in the network device using "ietf-node-tags" module defined in this document and fetch tagged data node instances and associated data path to the datastore node. The node tag information instruct the receiver to subscribe tagged data node (e.g., performance metric data nodes) using standard subscribed notification mechanism [RFC8639].

===== NOTE: '\ ' line wrapping per RFC 8792 =====

```
<?xml version="1.0" encoding="UTF-8"?>
  <t:module-tags
    xmlns:t="urn:ietf:params:xml:ns:yang:ietf-module-tags">
    <t:module>
      <t:name>ietf-interfaces</t:name>
      <s:node-tags
        xmlns:s="urn:ietf:params:xml:ns:yang:ietf-node-tags">
        <s:node>
          <s:id>/if:interfaces/if:interface/if:in-errors</s:id>
          <s:tags>
            <s:tag>ietf:metric</s:tag>
          </s:tags>
          <s:tags>
            <s:tag>ietf:loss</s:tag>
          </s:tags>
        </s:node>
      </s:node-tags>
    </t:module>
  </module-tags>
```

Figure 6: List of Available Target Objects

With node tag information returned, e.g., in the 'get-data' operation, the subscriber identifies tagged data node and associated data path to the datastore node and sends a standard establish-subscription RPC [[RFC8639](#)] to subscribe tagged data nodes that are interests to the client application from the publisher. The publisher returns specific data node types of operational state (e.g., in-errors statistics data) subscribed by the client as follows:

===== NOTE: '\ ' line wrapping per RFC 8792 =====

```
<netconf:rpc message-id="101"
  xmlns:netconf="urn:ietf:params:xml:ns:netconf:base:1.0">
  <establish-subscription
    xmlns="urn:ietf:params:xml:ns:yang:ietf-subscribed-notifica\
      tions"
    xmlns:yp="urn:ietf:params:xml:ns:yang:ietf-yang-push">
    <yp:datastore
      xmlns:ds="urn:ietf:params:xml:ns:yang:ietf-datastores">
      ds:operational
    </yp:datastore>
    <yp:datastore-xpath-filter
      xmlns:ex="https://example.com/sample-data/1.0">
      /if:interfaces/if:interface/if:statistics/if:in-errors
    </yp:datastore-xpath-filter>
    <yp:periodic>
      <yp:period>500</yp:period>
    </yp:periodic>
  </establish-subscription>
</netconf:rpc>
```

Appendix F. Changes between Revisions

Editorial Note (To be removed by RFC Editor)

v07 - v08

- *Make objective clearly, cover tags for both nodes in the schema tree and nodes in the data tree.
- *Document clearly which tags can be cached and how applications are supposed to resynchronize and pull in any update in section 3.
- *Clarify Instance level tag is not used to guide retrieval operations in section 3.
- *Distinguish Instance level tag from Metadata annotation in the introduction section.
- *Distinguish Schema Level tag from Instance level tag in the introduction section and section 3.
- *Schema Level tag used in xpath query has be clarified in section 3.
- *Other editorial changes.

v06 - v07

- *Update use case in section 3 to remove object and subobject concept and massive related words.
- *Change the title into Node Tags in YANG Modules.
- *Update Model Tag design in section 5.1 based on Balazs's comments.
- *Add Instance level tunnel tagging example in the Appendix.
- *Add 'type' parameter in the base model and add one more model extension example in the Appendix.
- *Consolidate opm-tag extension, metric-type extension and multi-source-tag extension into one generic yang extension.
- *Remove object tag and property tag.
- *Other Appendix Updates.

v05 - v06

- *Additional Editorial changes;
- *Use the folding defined in [RFC8792].

v04 - v05

- *Add user tag formatting clarification;
- *Provide guidance to the Designated Expert for evaluation of YANG node Tag registry and YANG node Tag prefix registry.
- *Update the figure 1 and figure 2 with additional tags.
- *Security section enhancement for user tag management.
- *Change data node name into name in the module.
- *Other Editorial changes to address Adrian's comments and comments during YANG docotor review.
- *Open issue: Are there any risks associated with an attacker adding or removing tags so that a requester gets the wrong data?

v03 - v04

- *Remove histogram metric type tag from metric type tags.

*Clarify the object tag and property tag, metric tag are mutual exclusive.

*Clarify to have two optional node tags (i.e., object tag and property tag) to indicate relationship between data nodes.

*Update targeted data node collection example.

v02 - v03

*Additional Editorial changes.

*Security section enhancement.

*Nits fixed.

v01 - v02

*Clarify the relation between data node, object tag, property tag and metric tag in figure 1 and figure 2 and related description;

*Change Metric Group into Metric Type in the YANG model;

*Add 5 metric types in section 7.2;

v00 - v01

*Merge node tag use case section into introduction section as a subsection;

*Add one glossary section;

*Clarify the relation between data node, object tag, property tag and metric tag in node Tags Use Case section;

*Add update to RFC8407 in the front page.

Authors' Addresses

Qin Wu
Huawei
101 Software Avenue, Yuhua District
Nanjing
Jiangsu, 210012
China

Email: bill.wu@huawei.com

Benoit Claise
Huawei

De Kleetlaan 6a b1
1831 Diegem
Belgium

Email: benoit.claise@huawei.com

Peng Liu
China Mobile
32 Xuanwumen West St, Xicheng District
Beijing

Email: liupengyjy@chinamobile.com

Zongpeng Du
China Mobile
32 Xuanwumen West St, Xicheng District
Beijing

Email: duzongpeng@chinamobile.com

Mohamed Boucadair
Orange
35000 Rennes
France

Email: mohamed.boucadair@orange.com