Internet Engineering Task Force (IETF)
Request for Comments: 8819
Updates: 8407
Category: Standards Track
ISSN: 2070-1721

C. Hopps
L. Berger
LabN Consulting, L.L.C.
D. Bogdanovic
Volta Networks

January 2021

YANG Module Tags

Abstract

This document provides for the association of tags with YANG modules. The expectation is for such tags to be used to help classify and organize modules. A method for defining, reading, and writing modules tags is provided. Tags may be registered and assigned during module definition, assigned by implementations, or dynamically defined and set by users. This document also provides guidance to future model writers; as such, this document updates RFC 8407.

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/rfc8819.

Table of Contents

1. Introduction

The use of tags for classification and organization is fairly ubiquitous not only within IETF protocols but in the internet itself (e.g., "#hashtags"). One benefit of using tags for organization over a rigid structure is that it is more flexible and can more easily adapt over time as technologies evolve. Tags can be usefully registered, but they can also serve as a non-registered mechanism available for users to define themselves. This document provides a mechanism to define tags and associate them with YANG modules in a flexible manner. In particular, tags may be registered as well as assigned during module definition, assigned by implementations, or dynamically defined and set by users.

This document defines a YANG module [RFC7950] that provides a list of module entries to allow for adding or removing tags as well as viewing the set of tags associated with a module.

This document defines an extension statement to indicate tags that SHOULD be added by the module implementation automatically (i.e., outside of configuration).

This document also defines an IANA registry for tag prefixes as well as a set of globally assigned tags.

Section 6 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 (NMDA) defined in [RFC8342].

1.1. Some Possible Use Cases for YANG Module Tags

During this document's development, there were requests for example uses of module tags. The following are a few example use cases for tags. This list is certainly not exhaustive.

One example use of tags would be to help filter different discrete categories of YANG modules supported by a device. For example, if modules are suitably tagged, then an XPath query can be used to list all of the vendor modules supported by a device.

Tags can also be used to help coordination when multiple, semi-independent clients are interacting with the same devices. For

example, one management client could mark that some modules should not be used because they have not been verified to behave correctly, so that other management clients avoid querying the data associated with those modules.

Tag classification is useful for users searching module repositories (e.g., YANG catalog). A query restricted to the 'ietf:routing' module tag could be used to return only the IETF YANG modules associated with routing. Without tags, a user would need to know the name of all the IETF routing protocol 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).

1.2. Conventions Used in This Document

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.

2. Tag Values

All tags SHOULD begin with a prefix indicating who owns their definition. An IANA registry (Section 7.1) is used to support registering tag prefixes. Currently, 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, and tabs.

Again, 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.

2.1. IETF Tags

An IETF tag is a tag that has the prefix "ietf:". All IETF tags are registered with IANA in a registry defined later in this document (Section 7.2).

2.2. 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; however, it is RECOMMENDED that the vendor include extra identification in the tag to avoid collisions, such as using the enterprise or organization name following the "vendor:" prefix (e.g., vendor:example.com:vendor-defined-classifier).

2.3. User Tags

A user tag is any tag that has the prefix "user:". These tags are defined by the user/administrator and are not meant to be registered. Users are not required to use the "user:" prefix; however, doing so is RECOMMENDED as it helps avoid collisions.

2.4. Reserved Tags

Any tag not starting with the prefix "ietf:", "vendor:", or "user:" is reserved for future use. These tag values are not invalid but

simply reserved in the context of specifications (e.g., RFCs).

3. Tag Management

Tags can become associated with a module in a number of ways. Tags may be defined and associated at module design time, at implementation time, or via user administrative control. As the main consumer of tags are users, users may also remove any tag, no matter how the tag became associated with a module.

3.1. Module Definition Tagging

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

If the module is defined in an IETF Standards Track document, the tags MUST be IETF tags (Section 2.1). Thus, new modules can drive the addition of new IETF tags to the IANA registry defined in Section 7.2, and the IANA registry can serve as a check against duplication.

3.2. Implementation Tagging

An implementation MAY include additional tags associated with a module. These tags SHOULD be IETF tags (i.e., registered) or vendor-specific tags.

3.3. User Tagging

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

4. Tags Module Structure

4.1. Tags Module Tree

The tree associated with the "ietf-module-tags" module follows. The meaning of the symbols can be found in [RFC8340].

```
module: ietf-module-tags
  +--rw module-tags
     +--rw module* [name]
        +--rw name          yang:yang-identifier
        +--rw tag*          tag
        +--rw masked-tag*   tag
```

Figure 1: YANG Module Tags Tree Diagram

4.2. YANG Module

```
<CODE BEGINS> file "ietf-module-tags@2021-01-04.yang"
module ietf-module-tags {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-module-tags";
  prefix tags;

  import ietf-yang-types {
    prefix yang;
  }

  organization
    "IETF NetMod Working Group (NetMod)";
  contact
```

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

   Author: Christian Hopps
           <mailto:chopps@chopps.org>

   Author: Lou Berger
           <mailto:lberger@labn.net>

   Author: Dean Bogdanovic
           <mailto:ivandean@gmail.com>";

description
  "This module describes a mechanism associating tags with YANG
   modules.  Tags may be IANA assigned or privately defined.

   Copyright (c) 2021 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 8819
   (https://www.rfc-editor.org/info/rfc8819); see the RFC itself
   for full legal notices.

   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.";

revision 2021-01-04 {
  description
    "Initial revision.";
  reference
    "RFC 8819: YANG Module Tags";
}

typedef tag {
  type string {
    length "1..max";
    pattern '[\S ]+';
  }
  description
    "A tag is a type of 'string' value that does not include
     carriage return, newline, or tab characters.  It SHOULD begin
     with a registered prefix; however, tags without a registered
     prefix SHOULD NOT be treated as invalid.";
}

extension module-tag {
  argument tag;
  description
    "The argument 'tag' is of type 'tag'.  This extension statement
     is used by module authors to indicate the tags that SHOULD be
     added automatically by the system.  As such, the origin of the
     value for the predefined tags should be set to 'system'
     [RFC8342].";
}

container module-tags {
```

```
      description
        "Contains the list of modules and their associated tags.";
      list module {
        key "name";
        description
          "A list of modules and their associated tags.";
        leaf name {
          type yang:yang-identifier;
          mandatory true;
          description
            "The YANG module name.";
        }
        leaf-list tag {
          type tag;
          description
            "Tags associated with the module.  See the IANA 'YANG
             Module Tag Prefixes' registry for reserved prefixes and
             the IANA 'IETF YANG Module Tags' registry for IETF tags.

             The 'operational' state [RFC8342] 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.";
        }
        leaf-list masked-tag {
          type tag;
          description
            "The list of tags that should not be associated with this
             module.  The user can remove (mask) tags from the
             operational state datastore [RFC8342] by adding them to
             this list.  It is not an error to add tags to this list
             that are not associated with the module, but they have no
             operational effect.";
        }
      }
    }
  }
  <CODE ENDS>
```

Figure 2: Module Tags Module

5. Other Classifications

It is worth noting that a different YANG module classification document exists [RFC8199]. That document only classifies modules in a logical manner and does not define tagging or any other mechanisms. It divides YANG modules into two categories (service or element) and then into one of three origins: standard, vendor, or user. It does provide a good way to discuss and identify modules in general. This document defines IETF tags to support the classification style described in [RFC8199].

6. Guidelines to Model Writers

This section updates [RFC8407].

6.1. Define Standard Tags

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

```
module example-module {
```

```
  namespace "https://example.com/yang/example";
  prefix "ex";
  //...
  import module-tags { prefix tags; }

  tags:module-tag "ietf:some-new-tag";
  tags:module-tag "ietf:some-other-tag";
  // ...
}
```

The module writer can use existing standard tags or use new tags defined in the model definition, as appropriate. For IETF standardized modules, new tags MUST be assigned in the IANA registry defined below, see Section 7.2.

7. IANA Considerations

7.1. YANG Module Tag Prefixes Registry

IANA has created the "YANG Module Tag Prefixes" subregistry in the "YANG Module Tags" registry.

This registry allocates tag prefixes. All YANG module tags SHOULD begin with one of the prefixes in this 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.

The initial values for this registry are as follows.

Prefix	Description	Reference	Assignee
ietf:	IETF tags allocated in the IANA "IETF YANG Module Tags" registry.	RFC 8819	IETF
vendor:	Non-registered tags allocated by the module implementer.	RFC 8819	IETF
user:	Non-registered tags allocated by and for the user.	RFC 8819	IETF

Table 1

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

7.2. IETF YANG Module Tags Registry

IANA has created the "IETF YANG Module Tags" subregistry within the "YANG Module Tags" registry . This registry appears below the "YANG Module Tag Prefixes" registry.

This registry 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. IANA assigned tags must conform to Net-Unicode as defined in [RFC5198], and they shall not

need normalization.

The allocation policy for this registry is IETF Review [RFC8126].

The initial values for this registry are as follows.

Tag	Description	Reference
ietf:network-element-class	Network element as defined in [RFC8199].	[RFC8199]
ietf:network-service-class	Network service as defined in [RFC8199].	[RFC8199]
ietf:sdo-defined-class	Module is defined by a standards organization.	[RFC8199]
ietf:vendor-defined-class	Module is defined by a vendor.	[RFC8199]
ietf:user-defined-class	Module is defined by the user.	[RFC8199]
ietf:hardware	Relates to hardware (e.g., inventory).	RFC 8819
ietf:software	Relates to software (e.g., installed OS).	RFC 8819
ietf:protocol	Represents a protocol (often combined with another tag to refine).	RFC 8819
ietf:qos	Relates to quality of service.	RFC 8819
ietf:network-service-app	Relates to a network service application (e.g., an NTP server, DNS server, DHCP server, etc.).	RFC 8819
ietf:system-management	Relates to system management (e.g., a system management protocol such as syslog, TACAC+, SNMP, NETCONF, etc.).	RFC 8819
ietf:oam	Relates to Operations, Administration, and Maintenance (e.g., BFD).	RFC 8819
ietf:routing	Relates to routing.	RFC 8819
ietf:security	Related to security.	RFC 8819
ietf:signaling	Relates to control-plane signaling.	RFC 8819
ietf:link-management	Relates to link	RFC 8819

	management.	

Table 2

7.3. Updates to the IETF XML Registry

This document registers a URI in the "IETF XML Registry" [RFC3688]. Following the format in [RFC3688], the following registrations have been made:

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

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

7.4. Updates to the YANG Module Names Registry

This document registers two YANG modules in the "YANG Module Names" registry [RFC6020]. Following the format in [RFC6020], the following registrations have been made:

name: ietf-module-tags
namespace: urn:ietf:params:xml:ns:yang:ietf-module-tags
prefix: tags
reference: RFC 8819

name: ietf-module-tags-state
namespace: urn:ietf:params:xml:ns:yang:ietf-module-tags-state
prefix: tags-s
reference: RFC 8819

8. Security Considerations

The YANG module defined in this memo is designed to be accessed via the NETCONF protocol [RFC6241]. The lowest NETCONF layer is the secure transport layer and the mandatory-to-implement secure transport is Secure Shell (SSH) [RFC6242].

This document adds the ability to associate tag metadata with YANG modules. This document does not define any actions based on these associations, and none are yet defined; therefore, it does not by itself introduce any new security considerations directly.

Users of the tag metadata may define various actions to be taken based on the tag metadata. 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.

9. References

9.1. Normative References

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

[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>.

[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>.

[RFC8199] Bogdanovic, D., Claise, B., and C. Moberg, "YANG Module Classification", RFC 8199, DOI 10.17487/RFC8199, July 2017, <https://www.rfc-editor.org/info/rfc8199>.

[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>.

[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>.

9.2. Informative References

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

[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>.

[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>.

[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>.

Appendix A. Examples

The following is a fictional NETCONF example result from a query of the module tags list. For the sake of brevity, only a few module results are shown.

```
<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-bfd</t:name>
      <t:tag>ietf:network-element-class</t:tag>
      <t:tag>ietf:oam</t:tag>
      <t:tag>ietf:protocol</t:tag>
      <t:tag>ietf:sdo-defined-class</t:tag>
```

```
      </t:module>
      <t:module>
        <t:name>ietf-isis</t:name>
        <t:tag>ietf:network-element-class</t:tag>
        <t:tag>ietf:protocol</t:tag>
        <t:tag>ietf:sdo-defined-class</t:tag>
        <t:tag>ietf:routing</t:tag>
      </t:module>
      <t:module>
        <t:name>ietf-ssh-server</t:name>
        <t:tag>ietf:network-element-class</t:tag>
        <t:tag>ietf:protocol</t:tag>
        <t:tag>ietf:sdo-defined-class</t:tag>
        <t:tag>ietf:system-management</t:tag>
      </t:module>
    </t:module-tags>
  </ns0:data>
```

Figure 3: Example NETCONF Query Output

Appendix B. 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-module-tags-state@2021-01-04.yang"
module ietf-module-tags-state {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-module-tags-state";
  prefix tags-s;

  import ietf-yang-types {
    prefix yang;
  }
  import ietf-module-tags {
    prefix tags;
  }

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

     Author: Christian Hopps
             <mailto:chopps@chopps.org>

     Author: Lou Berger
             <mailto:lberger@labn.net>

     Author: Dean Bogdanovic
             <mailto:ivandean@gmail.com>";

  description
    "This module describes a mechanism associating tags with YANG
     modules.  Tags may be IANA assigned or privately defined.

     This is a temporary non-NMDA module that is for use by
     implementations that don't yet support NMDA.

     Copyright (c) 2021 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 8819
       (https://www.rfc-editor.org/info/rfc8819); see the RFC itself
       for full legal notices.";

     revision 2021-01-04 {
       description
         "Initial revision.";
       reference
         "RFC 8819: YANG Module Tags";
     }

     container module-tags-state {
       config false;
       status deprecated;
       description
         "Contains the list of modules and their associated tags.";
       list module {
         key "name";
         status deprecated;
         description
           "A list of modules and their associated tags.";
         leaf name {
           type yang:yang-identifier;
           mandatory true;
           status deprecated;
           description
             "The YANG module name.";
         }
         leaf-list tag {
           type tags:tag;
           status deprecated;
           description
             "Tags associated with the module.  See the IANA 'YANG
              Module Tag Prefixes' registry for reserved prefixes and
              the IANA 'IETF YANG Module Tags' registry for IETF tags.

              The contents of this list is constructed using the
              following steps:

              1) System tags (i.e., tags of added by the system) are
              added.
              2) User-configured tags (i.e., tags added by
              configuration) are added.
              3) Any tag that is equal to a masked-tag present in the
              corresponding ietf-module-tags:module-tags:module-tag leaf
              list for this module is removed.";
         }
       }
     }
   }
   <CODE ENDS>
```

Figure 4: Non-NMDA Module Tags State Module

Acknowledgements

Special thanks to Robert Wilton for his help improving the introduction and providing the example use cases, as well as generating the non-NMDA module.

Authors' Addresses

Christian Hopps
LabN Consulting, L.L.C.

Email: chopps@chopps.org

Lou Berger
LabN Consulting, L.L.C.

Email: lberger@labn.net

Dean Bogdanovic
Volta Networks

Email: ivandean@gmail.com