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 a modules tags is provided. Tags may be standardized 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 RFC8407.

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 September 10, 2019.

Copyright Notice

Copyright (c) 2019 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.
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 standardized, but they can also serve as a non-standardized 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 standardized as well as assigned during module definition; assigned by implementations; or dynamically defined and set by users.

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

This document defines an extension statement to be used 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 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 [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 standardizing tag prefixes. Currently 3 prefixes are defined. No further structure is imposed by this document on the value following the standard 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 not specifying any structure on (i.e., restricting) the tag values on purpose. 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 Standard Tags

An IETF standard tag is a tag that has the prefix "ietf:". All IETF standard 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 standardized; 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 will never be standardized. 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 standardization. These tag values are not invalid, but simply reserved in the context of standardization.

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 Standard Tags (2.1). Thus, new modules can drive the addition of new standard 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 Standard 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
```

4.2. YANG Module

```<CODE BEGINS> file "ietf-module-tags@2019-03-09.yang"
module ietf-module-tags {
  yang-version 1.1;
  prefix tags;

  import ietf-yang-types {
    prefix yang;
  }

  organization
    "IETF NetMod Working Group (NetMod)";
  contact
    "WG Web: <https://tools.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
      <ivandean@gmail.com>
  
  // RFC Ed.: replace XXXX with actual RFC number and
  // remove this note.

  description
    "This module describes a mechanism associating tags with YANG
     modules. Tags may be IANA assigned or privately defined.
```
Copyright (c) 2018 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://tools.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 2019-03-09 {
    description
        "Initial revision.";
    reference "RFC XXXX: YANG Module Tags";
}

typedef tag {
    type string {
        length "1..max";
        pattern '([^\s ]+);'
    }
    description
        "A tag is a type 'string' value that does not include carriage return, newline or tab characters. It SHOULD begin with a standard prefix; however, tags without a standard 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 pre-defined 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 'YANG Module Tags' registry for IETF standard 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.";
      }
    }
}
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 standard tags to support [RFC8199] style classification.

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 {

  //...
  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 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 is asked to create a new registry "YANG Module Tag Prefixes" grouped under a new "Protocol" category named "YANG Module Tags".

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 initial values for this registry are as follows.

<table>
<thead>
<tr>
<th>Prefix</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>vendor:</td>
<td>Non-standardized tags allocated by the module implementer.</td>
</tr>
<tr>
<td>user:</td>
<td>Non-standardized tags allocated by and for the user.</td>
</tr>
</tbody>
</table>

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

7.2. YANG Module Tags Registry

IANA is asked to create a new registry "YANG Module Tags" grouped under a new "Protocol" category "YANG Module Tags". This registry should be included below "YANG Module Tag Prefixes" when listed on the same page.

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

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

The initial values for this registry are as follows.

<table>
<thead>
<tr>
<th>Tag</th>
<th>Description</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>ietf:network-element-class</td>
<td>[RFC8199] network element.</td>
<td>[RFC8199]</td>
</tr>
<tr>
<td>ietf:network-service-class</td>
<td>[RFC8199] network service.</td>
<td>[RFC8199]</td>
</tr>
<tr>
<td>ietf:sdo-defined-class</td>
<td>Module is defined by a standards organization.</td>
<td>[RFC8199]</td>
</tr>
<tr>
<td>ietf:vendor-defined-class</td>
<td>Module is defined by a standards organization.</td>
<td>[RFC8199]</td>
</tr>
<tr>
<td>ietf:user-defined-class</td>
<td>Module is defined by the user. [RFC8199]</td>
<td></td>
</tr>
<tr>
<td>------------------------</td>
<td>------------------------------------------</td>
<td></td>
</tr>
<tr>
<td>ietf:hardware</td>
<td>Relates to hardware (e.g., inventory). [This document]</td>
<td></td>
</tr>
<tr>
<td>ietf:software</td>
<td>Relates to software (e.g., installed OS). [This document]</td>
<td></td>
</tr>
<tr>
<td>ietf:protocol</td>
<td>Represents a protocol (often combined with another tag to refine). [This document]</td>
<td></td>
</tr>
<tr>
<td>ietf:qos</td>
<td>Relates to quality of service. [This document]</td>
<td></td>
</tr>
<tr>
<td>ietf:network-service-app</td>
<td>Relates to a network service application (e.g., an NTP server, DNS server, DHCP server, etc). [This document]</td>
<td></td>
</tr>
<tr>
<td>ietf:system-management</td>
<td>Relates to system management (e.g., a system management protocol such as syslog, TACAC+, SNMP, netconf, ...) [This document]</td>
<td></td>
</tr>
<tr>
<td>ietf:oam</td>
<td>Relates to Operations, Administration, and Maintenance (e.g., BFD). [This document]</td>
<td></td>
</tr>
<tr>
<td>ietf:routing</td>
<td>Relates to routing. [This document]</td>
<td></td>
</tr>
<tr>
<td>ietf:security</td>
<td>Related to security. [This document]</td>
<td></td>
</tr>
<tr>
<td>ietf:signaling</td>
<td>Relates to control plane signaling. [This document]</td>
<td></td>
</tr>
<tr>
<td>ietf:link-management</td>
<td>Relates to link management. [This document]</td>
<td></td>
</tr>
</tbody>
</table>
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 registration has been made:

URI:

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 one YANG module in the "YANG Module Names" registry [RFC6020]. Following the format in [RFC6020], the following registration has been made:

name:
    ietf-module-tags

namespace:

prefix:
    tags

reference:
    RFC XXXX (RFC Ed.: replace XXX with actual RFC number and remove this note.)

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 SSH [RFC6242].

This document adds the ability to associate tag meta-data with 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 tag-meta data may define various actions to be taken based on the 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.

9. References

9.1. Normative References


9.2. Informative References


Appendix A. Examples

The following is a fictional example result from a query of the module tags list. For the sake of brevity only a few module results are imagined.
<ns0:config xmlns:ns0="urn:ietf:params:xml:ns:netconf:base:1.0">
        <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:config>

Appendix B. Acknowledgements

Special thanks to Robert Wilton for his help improving the introduction and providing the example use cases.

Authors’ Addresses

Christian Hopps
LabN Consulting, L.L.C.

Email: chopps@chopps.org

Lou Berger
LabN Consulting, LLC.

Email: lberger@labn.net