

NETMOD Working Group Q.
Wu
Internet-Draft
Huawei
Updates: [8407](#) (if approved) B.
Claise
Intended status: Standards Track
Cisco
Expires: July 1, 2021 L.
Geng Z.
Du
China
Mobile
M.
Boucadair
Orange
December 28,
2020

**Self Describing Data Object Tags
draft-ietf-netmod-node-tags-00**

Abstract

This document defines a method to tag data objects associated with operation and management data in YANG Modules. This YANG data object tagging method can be used to classify data objects from different YANG modules and identify characteristics data. It also can provide input, instruction, indication to selection filter and filter queries of operational state on a server during a "pub/sub" service for YANG datastore updates. When the state of all subscriptions of a particular Subscriber to be fetched is huge, the amount of data to be streamed out to the destination can be greatly reduced and only targeted to the characteristics data. These data object tags may be registered as well as assigned during the module definition; assigned by implementations; or dynamically defined and set by users.

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 July 1, 2021.

Wu, et al.
1]

Expires July 1, 2021

[Page

Copyright Notice

Copyright (c) 2020 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 Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1.	Introduction	3
1.1.	Self Describing Data Object Tags Use Case	4
1.1.1.	Massive Data Object Collection	4
1.2.	Terminology	6
2.	Data Object Tag Values	6
2.1.	IETF Tags Prefix	7
2.2.	Vendor Tags Prefix	7
2.3.	User Tags Prefix	7
2.4.	Reserved Tags Prefix	7
3.	Data Object Tag Management	7
3.1.	Module Design Tagging	8
3.2.	Implementation Tagging	8
3.3.	User Tagging	9
4.	Data Object Tags Module Structure	9
4.1.	Data Object Tags Module Tree	9
5.	YANG Module	9
6.	Guidelines to Model Writers	12

12	6.1. Define Standard Tags
13	7. IANA Considerations
13	7.1. YANG Data Object Tag Prefixes Registry
14	7.2. IETF YANG Data Object Tags Registry
16	7.3. Updates to the IETF XML Registry
16	7.4. Updates to the YANG Module Names Registry
16	8. Security Considerations
17	9. Acknowledgements
17	10. References
17	10.1. Normative References
17	10.2. Informative References
18	Appendix A. NETCONF Example
19	Appendix B. Non-NMDA State Module
22	Appendix C. Targeted data object collection example

Authors' Addresses
[25](#)

[1.](#) Introduction

As described in [I.D-ietf-netmod-module-tags], the use of tags for classification and organization is fairly ubiquitous not only within IETF protocols, but in the internet itself (e.g., "#hashtags"). A module tag defined in [I.D-ietf-netmod-module-tags] is a string associated only with a module name at module level.

At the time of writing this document (2020), there are many data models that have been specified or are being specified by various different SDOs and Open Source community. They cover many of the networking protocols and techniques. However data objects defined by

these technology specific data models might represent a portion of fault, configuration, accounting, performance, security management categories information at different locations in various different ways, lack consistent classification criteria and representation for a specific service, feature or data source.

This document defines self-describing data object tags and associates them with data objects within YANG module, which

- o Provide dictionary meaning for specific targeted data objects;
- o Indicate relationship between data objects within the same YANG module or from different YANG modules;
- o Identify key performance metric data objects and the absolute XPath expression identifying the element path to the node;

The self describing data object tags can be used by the client to classify data objects from different YANG modules and identify characteristics data. In addition, it can provide input, instruction, indication to selection filter and filter queries of configuration or operational state on a server based on these data object tags, .e.g., return specific object type of operational state related to system-management. NETCONF clients can discover data objects with self describing data object tags supported by a NETCONF server via <get-schema> operation. The self describing data object tag capability can also be advertised via Capability Notification Model [I-D.netconf-notification-capabilities] by the NETCONF server or some place where offline document are kept. These data object tags may be registered as well as assigned during the module definition; assigned by implementations; or dynamically defined and set by users.

This document defines a YANG module [[RFC7950](#)] which augments module tag model and provides a list of data object entries to allow for adding or removing of self describing tags as well as viewing the set of self describing tags associated with specific data objects within YANG modules.

This document defines an extension statement to be used to indicate self describing 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.

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

[1.1.](#) Self Describing Data Object Tags Use Case

[1.1.1.](#) Massive Data Object Collection

Among data object tags, the opm (object, property subobject, metric subobject) tags can be used to tackle massive data objects collection and only capture YANG data objects associated with performance metrics data modelled with YANG (See Figure 1).

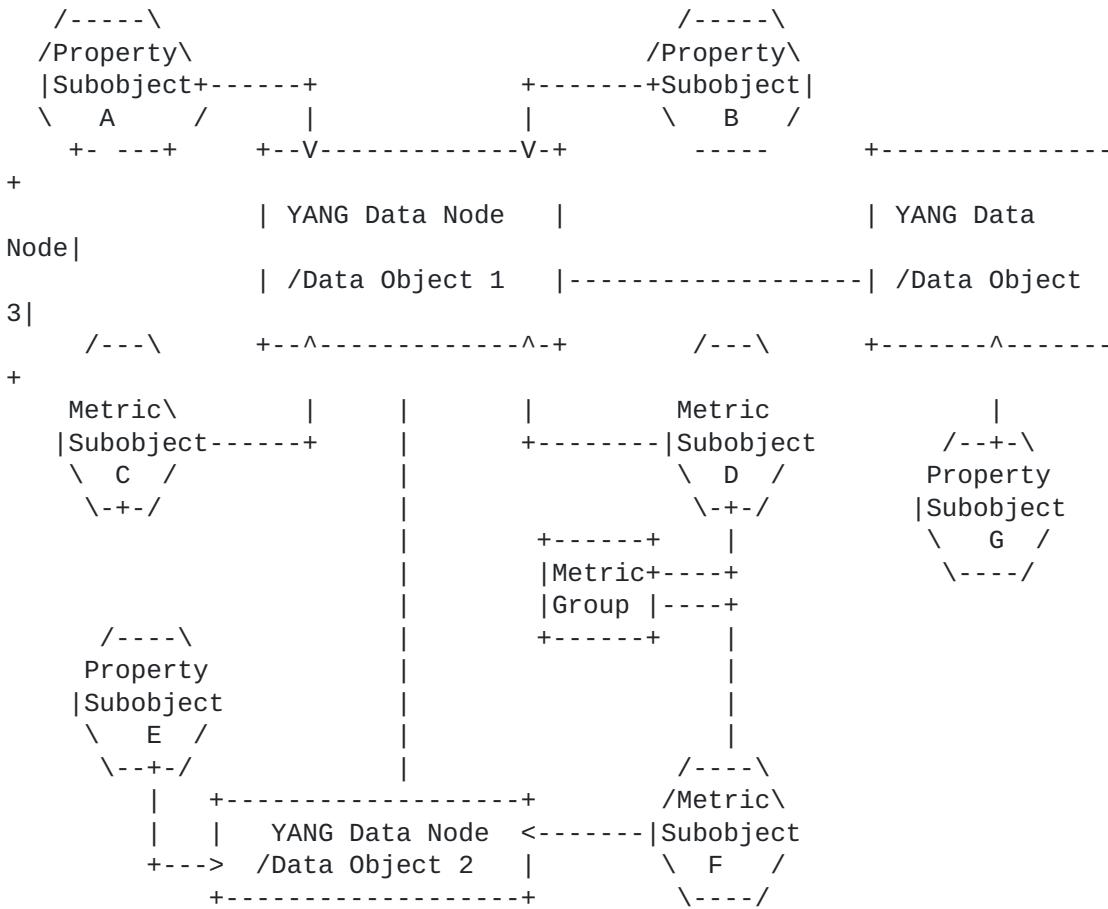


Figure 1: The Relation between Object, Property and Metric

In Figure 1, object can contain other objects called subobjects. Property and metric objects are both subobjects belonging to specific object. Both object and subobjects can be modeled as YANG data nodes [RFC7950]. Object can be one of container, leaf-list and list. Property subobject is a leaf node. Metric subobject can be one of container, leaf-list, list, leaf node. Object contains zero or many property subobjects, zero or many metric subobjects.

The use of opm tags would be to help filter discrete categories of YANG data objects scattered across the same or different YANG modules

supported by a device and capture all network performance data or all property data in the single view of the truth (see Figure 2). In Figure 2, tunnel-svc data object is a container node in the tunnel-pm module and can be seen as the root object for property subobjects (e.g., tunnel-svc/create-time) and metric subobjects (e.g., tunnel-svc/avg-latency). Name, create-time, modified-time are property

subobjects under tunnel-svc container. Avg-latency, packet loss are metric subobjects under tunnel-svc container node. In addition, not all metric subobjects need to be tagged, e.g., only specific category (e.g., loss related) metric subobjects need to be tagged with metric-group tag which can further reduce amount data to be fetched.

Data	Object	Property	Metric	Module
Object	Tag	Subobject	Subobject	Name
		Tag	Tag	
	ietf:			
tunnel-svc	object			tunnel-
pm		ietf:		
tunnel-svc/name		property		tunnel-
pm		ietf:		
tunnel-svc/create-time		property		tunnel-
pm		ietf:		
tunnel-svc/modified-time		property		tunnel-
pm				
tunnel-svc/avg-latency			ietf:	tunnel-
pm			metric	
tunnel-svc/packet-loss			ietf:	tunnel-
pm			metric	
tunnel-svc/min-latency			ietf:	tunnel-
pm			metric	
tunnel-svc/ max-latency			ietf:	tunnel-
pm			metric	

Figure 2: Example of OPM Tags Used in the YANG Module

If data objects in these YANG modules are suitably tagged and learnt by the client from a live server, the client can retrieve paths to all targeted data objects and then use an XPath query defined

[\[RFC8639\]](#) [\[RFC8641\]](#) to list all tagged data objects which reflect network characteristics

1.2. 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.

2. Data Object Tag Values

All data object tags SHOULD begin with a prefix indicating who owns their definition. An IANA registry ([Section 7.1](#)) is used to support registering data object tag prefixes. Currently 3 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. Therefore, 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 Prefix

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

2.2. Vendor Tags Prefix

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:vendor-defined-classifier).

2.3. User Tags Prefix

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 prefix collisions.

2.4. Reserved Tags Prefix

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. Data Object Tag Management

Tags can become associated with a data object within YANG module in a number of ways. Tags may be defined and associated at the module design time, at implementation time without the need of live server, or via user administrative control. As the main consumer of data object tags are users, users may also remove any tag from a live server, no matter how the tag became associated with a data object within a YANG module.

3.1. Module Design Tagging

A data object definition MAY indicate a set of data object tags to be added by the module implementer. These design time tags are indicated using a set of extension statements which include:

opm-tag extension statement: Classify management and operation data into object, property subobject and metric subobject three categories. Object can contain other objects called subobjects. Property and metric objects are both subobjects belonging to specific object. Both object and subobjects can be modeled as data nodes [[RFC7950](#)]. Object can be one of container, leaf-list and list. Property subobject is a leaf node. Metric subobject can be one of container, leaf-list, list, leaf. Object contains zero or many property subobjects, zero or many metric subobjects. See opm-tag example in Figure 2 and Figure 3.

metric-group extension statement: Provide metric subobjects classification (e.g., loss, jitter, delay) within the YANG module.

multi-source-tag extension statement: Identify multi-source aggregation type (e.g., aggregated, non-aggregated) related to metric subobject. 'aggregated' multi-source aggregation type allows a large number of measurements on metric subobjects from different sources of the same type (e.g., line card, each subinterface of aggregated Ethernet interface) being combined into aggregated statistics and report as one metric subobject. 'non-aggregated' multi-source aggregation type allows measurement from each source of the same type (e.g., line card, each subinterface of aggregated Ethernet interface) be reported separately.

Among these extension statements, the metric-group, multi-source-tag extension statements are context information related and can be used to correlate data object from the different modules.

If the data node is defined in an IETF standards track document, the data object tags MUST be IETF Tags (2.1). Thus, new data object can drive the addition of new IETF tags to the IANA registry defined in [Section 7](#), and the IANA registry can serve as a check against duplication.

3.2. Implementation Tagging

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

[3.3.](#) User Tagging

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

[4.](#) Data Object Tags Module Structure

[4.1.](#) Data Object Tags Module Tree

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

```
module: ietf-data-object-tags
  augment /tags:module-tags/tags:module:
    +--rw data-object-tags
      +--rw data-object* [object-name]
        +--rw object-name      nacm:node-instance-identifier
        +--rw tag*             tags:tag
        +--rw masked-tag*      tags:tag
```

[5.](#) YANG Module

```
<CODE BEGINS> file "ietf-data-object-tags@2019-05-03.yang"
module ietf-data-object-tags {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-data-object-tags";
  prefix ntags;

  import ietf-netconf-acm {
    prefix nacm;
  }
  import ietf-module-tags {
    prefix tags;
  }

  organization
    "IETF NetMod Working Group (NetMod)";
  contact
    "WG Web: <https://tools.ietf.org/wg/netmod/>
    WG List: <mailto:netmod@ietf.org>
    Editor: Qin Wu <mailto:bill.wu@huawei.com>
    Editor: Benoit Claise <mailto:bclaise@cisco.com>
    Editor: Liang Geng <mailto:gengliang@chinamobile.com>
    Editor: Zongpeng Du <mailto:duzongpeng@chinamobile.com>
```


Editor: Mohamed Boucadair <mailto:mohamed.boucadair@orange.com>;
description

"This module describes a mechanism associating self-describing tags with YANG data object within YANG modules. Tags may be IANA assigned or privately defined.

Copyright (c) 2020 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.";

```
revision 2019-05-03 {  
  description  
    "Initial revision.";  
  reference  
    "RFC XXXX: YANG Data Object Tags";  
}
```

```
extension opm-tag {  
  argument tag;  
  description  
    "The argument 'tag' is of type 'tag'. This extension statement  
    is used by module authors to indicate the opm tags that SHOULD  
be  
    added automatically by the system. Opm Tag is used to classify  
operation and management data into object, property subobject,  
and metric  
    subobject three categories. Object can contain other objects  
called subobjects.  
    Property and metric objects are both subobjects belonging to  
specific object.  
    Both object and subobjects can be modeled as data nodes. Object  
can be one of  
    container, leaf-list and list. Property subobject is a leaf  
node. Metric subobject  
    can be one of container, leaf-list, list, leaf. Object contains  
zero or many  
    property subobjects, zero or many metric subobjects. As such the  
origin of the  
    value for the pre-defined tags should be set to  
'system' [RFC8342].";  
}
```

```
extension metric-group {  
  argument tag;  
  description  
    "The argument 'tag' is of type 'tag'.The metric-group can be  
    used to provide metric subobject classification
```

```
    (e.g., loss, jitter, packet loss) within the YANG module.";
}

extension multi-source-tag {
  argument tag;
  description
    "The argument 'tag' is of type 'tag'.The multi-source-tag can be
    used to identify multi-source aggregation type (e.g.,
    aggregated,
    non-aggregated) related to metric subobject.

    'aggregated' multi-source aggregation type allows a large number
    of
    measurements on metric subobjects from different sources of the
    same
    type (e.g.,line card, each subinterface of aggregated Ethernet
    interface)
    being combined into aggregated statistics and report as one
    metric subobject
    value. 'non-aggregated' multi-source aggregation type allows
    measurement from
    each source of the same type (e.g.,line card, each subinterface
    of aggregated
    Ethernet interface) be reported separately.";
}

augment "/tags:module-tags/tags:module" {
  description
    "Augment the Module Tags module with data object tag attributes";
  container data-object-tags {
    description
      "Contains the list of data objects and their associated data
    object tags";
    list data-object {
      key "object-name";
      description
        "A list of data objects and their associated data object
    tags";
      leaf object-name {
        type nacm:node-instance-identifier;
        mandatory true;
        description
          "The YANG data object name.";
      }
      leaf-list tag {
        type tags:tag;
        description
          "Tags associated with the data object within the YANG
    module. See
    reserved
    the IANA 'YANG Data Object Tag Prefixes' registry for
    prefixes and the IANA'IETF YANG Data Object 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 tags:tag;
        description
            "The list of tags that should not be associated with the
data
            object within the YANG 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 data object within YANG
module,
            but they have no operational effect.";
    }
}
}
}
}
}
<CODE ENDS>
```

6. Guidelines to Model Writers

This section updates [[RFC8407](#)].

6.1. Define Standard Tags

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


```
module example-module-A {
  //...
  import ietf-data-node-tags { prefix ntags; }
  container top {
    ntags:opm-tag "ietf:object";
    list X {
      leaf foo {
        ntags:opm-tag "ietf:property";
      }
    }
  }
  container Y {
    leaf bar {
      ntags:opm-tag "ietf:metric";
    }
  }
}
// ...
}
```

Figure 3: Data object tag example

The module writer can use existing standard data object tags, or use new data object tags defined in the data object definition, as appropriate. For IETF standardized modules, new data object tags MUST be assigned in the IANA registry defined below, see Section [Section 7.2](#).

7. IANA Considerations

7.1. YANG Data Object Tag Prefixes Registry

IANA is asked to create a new registry "YANG Data Object Tag Prefixes" grouped under a new "Protocol" category named "YANG Data Object Tag Prefixes".

This registry allocates tag prefixes. All YANG Data Object 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 Data Object Tags registry document	[This document]	IETF
vendor:	Non-registered tags allocated by the module implementer.	[This document]	IETF
user:	Non-registered tags allocated by and for the user.	[This document]	IETF

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

7.2. IETF YANG Data Object Tags Registry

IANA is asked to create 3 new registries "IETF OPM Tags", "IETF Metric Group Tags", "IETF Multiple Source Tags" grouped under a new "Protocol" category. These 3 registries should be included below "YANG Data Object Tag Prefixes" when listed on the same page.

3 registries allocate 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 these three registries is IETF Review [[RFC8126](#)].

The initial values for these three registries are as follows.

Wu, et al.
14]

Expires July 1, 2021

[Page

OPM Tag	Description	Reference
ietf:object	Represent specific object type(e.g., interfaces).	[This document]
ietf:property	Represent a property subobject (e.g.,ifindex) associated with specific object (e.g.,interfaces).	[This document]
ietf:metric	Represent metric subobject (e.g., ifstatistics) associated with specific object(e.g.,interfaces)	[This document]
Metric Group Tag	Description	Reference
ietf:delay	Represent the metric group which metric subobjects belong to (i.e., delay)	[This document]
ietf:jitter	Represent the metric group which metric subobjects	[This document]

	belong to (i.e., jitter)	
ietf:loss	Represent the metric group	[This
	which metric subobjects	document]
	belong to (i.e., loss)	
+-----+-----+-----		
+-----+-----+-----		
Multiple Source Tag	Description	Reference
+-----+-----+-----		
ietf:non-agg	Relate to multiple source	[This
	aggregation type(i.e.,	document]
	aggregated statistics)	
ietf:agg	Relate to multiple source	[This
	aggregation type(i.e., non	document]
	aggregated statistics)	
+-----+-----+-----		
+-----+-----+-----		

Each YANG data object can have one opm tag, zero or one metric-group tag, zero or one multi-source tag.

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:
    urn:ietf:params:xml:ns:yang:ietf-data-object-tags
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-data-object-tags
namespace:
    urn:ietf:params:xml:ns:yang:ietf-data-object-tags
prefix:
    ntags
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 data object tag meta-data

with data object 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 data object tag meta-data may define various actions to be taken based on the data object 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. 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, Peng Liu, YingZhen Qu, Boyuan Yan.

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

10.2. 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>>.

- [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>>.
- [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>>.

[Appendix A](#). NETCONF Example

The following is a fictional NETCONF example result from a query of the data object tags list. For the sake of brevity only a few module and associated data object results are imagined.


```

<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:data-object-tags xmlns:s="urn:ietf:params:xml:ns:yang:ietf-
data-object-tags">
        <s:data-object>
          <s:object-name>/if:interfaces/if:interface</s:object-name>
          <s:tag>ietf:object</s:tag>
        </s:data-object>
        <s:data-object>
          <s:object-name>/if:interfaces/if:interface/if:last-change</
s:object-name>
          <s:tag>ietf:property</s:tag>
        </s:data-object>
        <s:data-object>
          <s:object-name>
            /if:interfaces/if:interface/if:statistics/if:in-errors
          </s:object-name>
          <s:tag>ietf:metric</s:tag>
        </s:data-object>
      </s:data-object-tags>
    </t:module>
    <t:module>
      <t:name>ietf-ip</t:name>
      <s:data-object-tags xmlns:s="urn:ietf:params:xml:ns:yang:ietf-
data-object-tags">
        <s:data-object>
          <s:object-name>/if:interfaces/if:interface/ip:ipv4</s:object-
name>
          <s:tag>ietf:object</s:tag>
        </s:data-object>
        <s:data-object>
          <s:object-name>/if:interfaces/if:interface/ip:ipv4/ip:enable</
s:object-name>
          <s:tag>ietf:property</s:tag>
        </s:data-object>
        <s:data-object>
          <s:object-name>/if:interfaces/if:interface/ip:ipv4/ip:mtu</
s:object-name>
          <s:tag>ietf:metric</s:tag>
        </s:data-object>
      </s:data-object-tags>
    </t:module>
  </t:module-tags>
</ns0:data>

```

[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-data-object-tags-state@2019-05-03.yang"  
module ietf-data-object-tags-state {
```

Wu, et al.
19]

Expires July 1, 2021

[Page

```
yang-version 1.1;
namespace "urn:ietf:params:xml:ns:yang:ietf-data-object-tags-state";
prefix ntags-s;

import ietf-netconf-acm {
  prefix nacm;
}
import ietf-module-tags {
  prefix tags;
}
organization
  "IETF NetMod Working Group (NetMod)";
contact
  "WG Web: <https://tools.ietf.org/wg/netmod/>
  WG List: <mailto:netmod@ietf.org>
  Editor: Qin Wu <mailto:bill.wu@huawei.com>
  Editor: Benoit Claise <mailto:bclaise@cisco.com>
  Editor: Liang Geng <mailto:gengliang@chinamobile.com>
  Editor: Zongpeng Du <mailto:duzongpeng@chinamobile.com>";
description
  "This module describes a mechanism associating self-describing
  tags with YANG data object within YANG modules. Tags may be IANA
  assigned or privately defined.

  Copyright (c) 2020 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).https://tools.ietf.org/html/rfcXXXX); see the RFC itself for
  full legal notices.";

revision 2019-05-03 {
  description
    "Initial revision.";
  reference
    "RFC XXXX: YANG Data Object Tags";
}

extension opm-tag {
  argument tag;
  description
```


"The argument 'tag' is of type 'tag'. This extension statement is used by module authors to indicate the opm tags that SHOULD be added automatically by the system. Opm Tag is used to classify operation and management data into object, property subobject, and metric subobject three categories. Object can contain other objects called subobjects. Property and metric objects are both subobjects belonging to specific object. Both object and subobjects can be modeled as data nodes. Object can be one of container, leaf-list and list. Property subobject is a leaf node. Metric subobject can be one of container, leaf-list, list, leaf. Object contains zero or many property subobjects, zero or many metric subobjects. As such the origin of the value for the pre-defined tags should be set to 'system'[RFC8342].";

```
}
extension metric-group {
  argument tag;
  description
    "The argument 'tag' is of type 'tag'.The metric-group can be
    used to provide metric subobject classification
    (e.g., loss, jitter, packet loss)within the YANG module.";
}
extension multi-source-tag {
  argument tag;
  description
    "The argument 'tag' is of type 'tag'.The multi-source-tag can be
    used to identify multi-source aggregation type (e.g.,
    aggregated,
    non-aggregated) related to metric subobject.

    'aggregated' multi-source aggregation type allows a large number
    of
    measurements on metric subobjects from different sources of the
    same
    type (e.g.,line card, each subinterface of aggregated Ethernet
    interface)
    being combined into aggregated statistics and report as one
    metric subobject
    value. 'non-aggregated' multi-source aggregation type allows
    measurement from
    each source of the same type (e.g.,line card, each subinterface
    of aggregated
    Ethernet interface) be reported separately.";
}

augment "/tags:module-tags/tags:module" {
  description
```

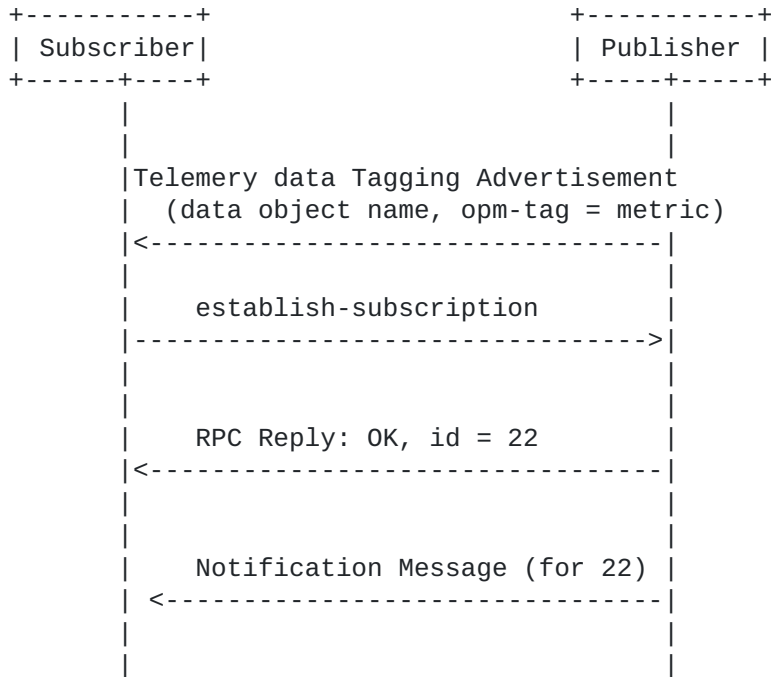
```
    "Augment the Module Tags module with data object tag attributes";
container data-object-tags {
    config false;
    status deprecated;
    description
        "Contains the list of data objects and their associated self
describing tags";
    list data-object {
        key "object-name";
        status deprecated;
        description
            "A list of data objects and their associated self describing
tags";
```


production, the actual values of "id" might not be small integers.

Wu, et al.
22]

Expires July 1, 2021

[Page



The publisher advertises telemetry data object capability to the subscriber to instruct the receiver to subscribe tagged data object (e.g., performance metric data object) using standard subscribed notification mechanism [[RFC8639](#)].

The following XML example [W3C.REC-xml-20081126] illustrates the advertisement of the list of available target objects using YANG instance file format [I-D.ietf-netmod-yang-instance-file-format]:


```
<?xml version="1.0" encoding="UTF-8"?>
<instance-data-set xmlns=\
  "urn:ietf:params:xml:ns:yang:ietf-yang-instance-data">
  <name>acme-router-notification-capabilities</name>
  <content-schema>
    <module>ietf-system-capabilities@2020-03-23</module>
    <module>ietf-notification-capabilities@2020-03-23</module>
    <module>ietf-data-export-capabilities@2020-03-23</module>
  </content-schema>
  <!-- revision date, contact, etc. -->
  <description>Defines the notification capabilities of an acme-router.
    The router only has running, and operational datastores.
    Every change can be reported on-change from running, but
    only config=true nodes and some config=false data from operational.
    Statistics are not reported based on timer based trigger and
counter
    threshold based trigger.
  </description>
  <content-data>
    <system-capabilities \
      xmlns="urn:ietf:params:xml:ns:yang:ietf-system-capabilities" \
      xmlns:inc=\
        "urn:ietf:params:xml:ns:yang:ietf-notification-capabilities" \
      xmlns:ds="urn:ietf:params:xml:ns:yang:ietf-datastores">
      <datastore-capabilities>
        <datastore>ds:operational</datastore>
        <per-node-capabilities>
          <node-selector>\
            /if:interfaces/if:interface/if:statistics/if:in-errors\
          </node-selector>
          <sec:self-describing-capabilities>
            <sec:opm-tag>metric</sec:opm-tag>
            <sec:metric-group>loss</sec:metric-group>
          </sec:self-describing-capabilities>
        </per-node-capabilities>
      </datastore-capabilities>
    </system-capabilities>
  </content-data>
</instance-data-set>
```

With telemetry data tagging information carried in the Telemetry data

Tagging Advertisement, the subscriber identifies targeted data object

and associated data path to the datastore node and sends a standard establish-subscription RPC [[RFC8639](#)] to subscribe tagged data objects

that are interests to the client application from the publisher.


```
<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-
notifications"
    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>
```

The publisher returns specific object type of operational state (e.g., in-errors statistics data) subscribed by the client.

Authors' Addresses

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

Email: bill.wu@huawei.com

Benoit Claise
Cisco
De Kleetlaan 6a b1
Diegem 1831
Belgium

Email: bclaise@cisco.com

Liang Geng
China Mobile
32 Xuanwumen West St, Xicheng District
Beijing 10053

Email: gengliangyjy@hotmail.com

Internet-Draft
2020

Data Object Tags

December

Zongpeng Du
China Mobile
32 Xuanwumen West St, Xicheng District
Beijing 10053

Email: duzongpeng@chinamobile.com

Mohamed Boucadair
Orange
Rennes 35000
France

Email: mohamed.boucadair@orange.com

