

NETMOD Working Group
Internet-Draft
Updates: [8407](#) (if approved)
Intended status: Standards Track
Expires: 31 October 2022

Q. Wu
B. Claise
Huawei
P. Liu
Z. Du
China Mobile
M. Boucadair
Orange
29 April 2022

Data Node Tags in YANG Modules
draft-ietf-netmod-node-tags-07

Abstract

This document defines a method to tag data nodes that are associated with operation and management data in YANG modules. This method for tagging YANG data nodes is meant to be used for classifying data nodes or instance 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 31 October 2022.

Copyright Notice

Copyright (c) 2022 IETF Trust and the persons identified as the document authors. All rights reserved.

Internet-Draft

YANG Node Tags

April 2022

This document is subject to [BCP 78](#) and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Revised BSD License text as described in Section 4.e of the [Trust Legal Provisions](#) and are provided without warranty as described in the Revised BSD License.

Table of Contents

1.	Introduction	3
2.	Terminology	4
3.	Data Classification and Fetching using Data Node Tags	4
4.	Data Node Tag Values	7
4.1.	IETF Tags	7
4.2.	Vendor Tags	7
4.3.	User Tags	7
4.4.	Reserved Tags	8
5.	Data Node Tag Management	8
5.1.	Module Design Tagging	8
5.2.	Implementation Tagging	10
5.3.	User Tagging	10
6.	Data Node Tags Module Structure	10
6.1.	Data Node Tags Module Tree	10
7.	YANG Module	10
8.	Guidelines to Model Writers	14
8.1.	Define Standard Tags	14
9.	IANA Considerations	15
9.1.	YANG Data Object Tag Prefixes Registry	15
9.2.	IETF YANG Data Object Tags Registry	16
9.3.	Updates to the IETF XML Registry	18
9.4.	Updates to the YANG Module Names Registry	18
10.	Security Considerations	18
11.	Acknowledgements	19
12.	Contributors	19
13.	References	19
13.1.	Normative References	19
13.2.	Informative References	20
Appendix A.	Example: Additional Auxiliary Data Property Information	22
Appendix B.	Instance Level Tunnel Tagging Example	22
Appendix C.	NETCONF Example	24

Appendix D.	Non-NMDA State Module	25
Appendix E.	Targeted Data Fetching Example	28
Appendix F.	Changes between Revisions	30
Authors' Addresses	32

[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 categorised in various different ways. Furthermore, there is no consistent classification criteria or representations for a specific service, feature, or data source.

This document defines data node tags and shows how they may be associated with data 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 key performance metric related data nodes and the absolute XPath expression identifying the element path to the nodes.

The data node tags can be used by a NETCONF [[RFC6241](#)] or RESTCONF [[RFC8040](#)] client to classify data nodes of instance of these data nodes from different YANG modules and identify characteristic data. In addition, these tags can provide input, instructions, or indications to selection filters and filter queries of configuration

or operational state on a server based on these data node tags (e.g., return specific data containing operational state related to performance management). NETCONF clients can discover data nodes or instances of data nodes with data node tags supported by a NETCONF server by means of the <get-schema> operation ([Section 3.1 of \[RFC6022\]](#)). The data node tag information can also be queried using the model defined in [Section 6.1](#). Similar to YANG module tags defined in [\[RFC8819\]](#), these data node tags may be registered or assigned during the module definition, assigned by implementations, or dynamically defined and set by users.

This document defines a YANG module [\[RFC7950\]](#) that augments the module tag model [\[RFC8819\]](#) and provides a list of data node instance entries to add or remove data node tags as well as to view the set of data node tags associated with specific data nodes or instance of data nodes within YANG modules.

This document defines three extension statements to indicate data node 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 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 [\[RFC8342\]](#).

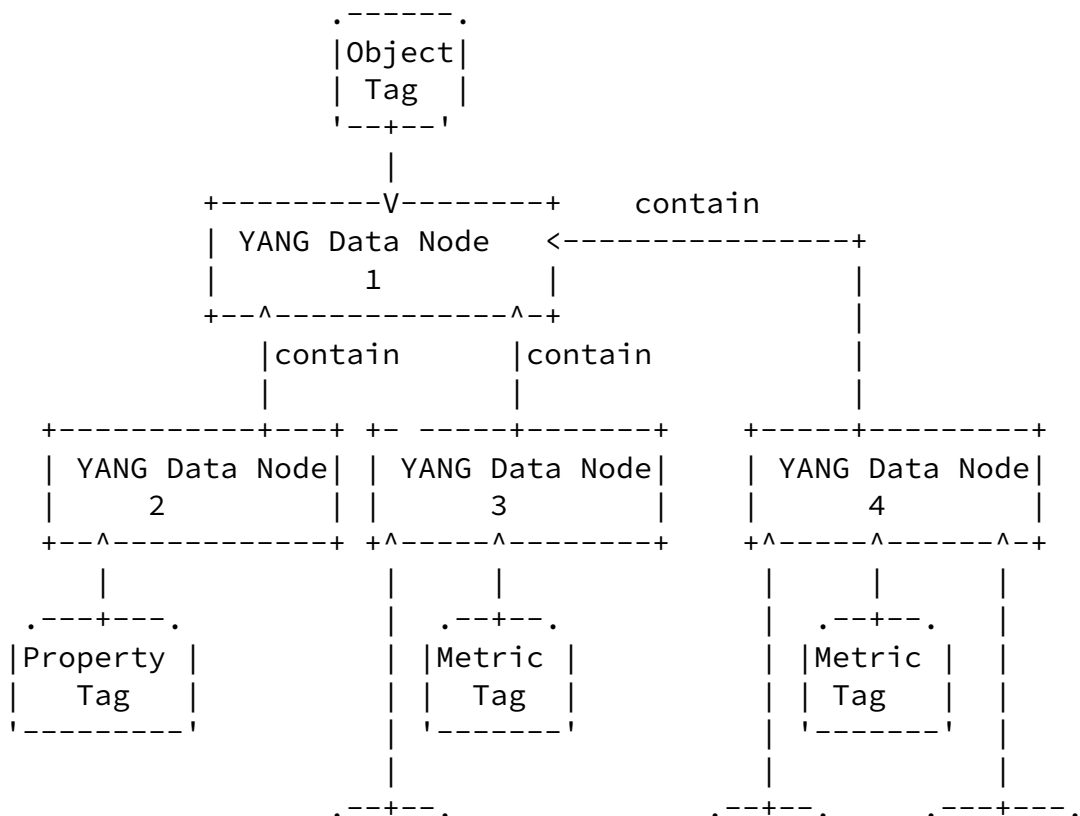
[2](#). Terminology

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

3. Data Classification and Fetching using Data Node Tags

Among data node tags, the 'opm' (object, property, metric) tags can be used to classify collected data, indicate relationships between data nodes, and capture YANG-modelled performance metrics data associated with data nodes or instances of data nodes. An example is depicted in Figure 1.



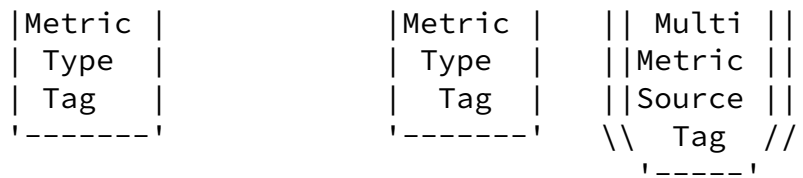


Figure 1: The Relation between Object, Property, and Metric

In Figure 1.

Data nodes that contain other data nodes can be one of type 'container', 'leaf-list', or 'list' and are tagged with the 'object' tag.

A Data node tagged with the 'property' tag is a 'leaf' node.

Data nodes tagged with the 'metric' tag can be one of 'container', 'leaf-list', 'list', or 'leaf' data node.

A data node may be associated with one single 'object' tag, or one single 'property' tag, or one single 'metric' tag. The data node tagged with the 'metric' tag also can have one or multiple MetricType tags and/or one single multi-source tag.

The use of 'opm' tags is meant to help filter discrete categories of YANG data objects scattered across the same or different YANG modules that are supported by a device and capture all network performance data or all property data in a single view of the data. In the example shown in Figure 2, the 'tunnel-svc' data node is a list node defined in a 'example-tunnel-pm' module and can be seen as the root object for property tagged data node (e.g., 'tunnel-svc'/'create-time') and metric tagged data node (e.g., 'tunnel-svc'/'avg-latency'). The 'name', 'create-time', and 'modified-time' are property tagged data node under 'tunnel-svc' list. The 'avg-latency' and 'packet-loss' metrics are metric tagged data nodes under 'tunnel-svc' list node. Consider the 'tunnel-svc' data node and the 'tunnel-svc/name' data node as an example: the 'tunnel-svc' data node has one single 'object' tag (i.e., 'ietf:object'), while the 'tunnel-svc/name' data node has one single 'property' data node tag (i.e.,

'ietf:property'). In addition, not all metric data node need to be tagged (e.g., define specific categories, such as loss-related metric data nodes need to be tagged with a metric-type tag which can further reduce amount data to be fetched).

Data Node	Object Tag	Property Tag	Metric Tag	Multi-Source Tag
tunnel-svc	ietf:object			
tunnel-svc/name		ietf:property		
tunnel-svc/create-time		ietf:property		
tunnel-svc/modified-time		ietf:property		
tunnel-svc/avg-latency			ietf:metric	non-agg
tunnel-svc/packet-loss			ietf:metric	non-agg
tunnel-svc/min-latency			ietf:metric	non-agg
tunnel-svc/max-latency			ietf:metric	non-agg

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

If data objects in YANG modules are adequately tagged and learnt by the client from a server, the client can retrieve paths to all targeted data nodes and then use an XPath query defined in [\[RFC8639\]](#) [RFC8641] to list all tagged data nodes which reflect the network characteristics.

4. Data Node Tag Values

All data 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 data 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 data node tag that has the prefix "ietf:".

All IETF data 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.](#) Data 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 data 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 data node tags to be added by a module's implementer. These design time tags are indicated using a set of extension statements which include:

opm-tag extension statement: Classifies management and operation data into object, property, and metric three categories. Three values (object, property and metric) are assigned to the 'opm-tag' tag.

Data nodes that contain other data nodes can be one of type 'container', 'leaf-list', and 'list' and are tagged with the 'object' tag value. A data node tagged with the 'property' tag value is a 'leaf' node. Data node tagged with the 'metric' tag value can be one of 'container', 'leaf-list', 'list', or 'leaf' data node. A data node contains one single 'object' tag, one single 'property' tag, or one single 'metric' tag. Both 'object' tag value and 'property' tag value are not inherited down the containment hierarchy, e.g., if a container is marked with a 'object' tag value, all its contained leaves don't inherit the tag value. The 'metric' tag value is inherited down the containment hierarchy if Data nodes tagged with the 'metric' tag is one of 'container', 'leaf-list', 'list'.

A data node tagged with the 'metric' tag also can have one or multiple Metric type tag and/or one single multi-source tag. See the examples depicted in Figure 2 and Figure 4.

metric-type extension statement: Provides metric related data nodes classifications (e.g., loss, jitter, delay, counter, gauge, summary, unknown) for data nodes tagged with the 'metric' tag. Data nodes tagged with the 'metric-type' tag can be one of 'container', 'leaf-list', 'list', or 'leaf' data node. The 'metric-type' tag is inherited down the containment hierarchy if Data nodes tagged with the 'metric-type' tag is one of 'container', 'leaf-list', 'list'. Figure 6 provides a list of possible values for the 'metric-type' tag.

multi-source-tag extension statement: Identifies multi-source aggregation type for data nodes tagged with the 'metric' tag. Two values (i.e., aggregated, non-aggregated) are assigned to 'multi-source-tag' tag.

The '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) to be combined into aggregated statistics and report as one metric subobject.

The '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) to be reported separately.

Data nodes tagged with the 'multi-source-tag' tag can be one of 'container', 'leaf-list', 'list', or 'leaf' data node. The 'multi-source-tag' tag is inherited down the containment hierarchy if Data nodes tagged with the 'multi-source-tag' tag is one of 'container', 'leaf-list', 'list'.

Among these extension statements, the 'metric-type' and 'multi-source-tag' extension statements are context information that can be used to correlate data nodes from the different modules.

If the data node is defined in an IETF Standards Track document, the data 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

data 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 data node tag from the operational datastore, the user adds a matching "masked-tag" entry for a given data node within the 'ietf-data-node-tags' module.

[6.](#) Data Node Tags Module Structure

[6.1.](#) Data Node Tags Module Tree

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

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

Figure 3: YANG Module Data Node Tags Tree Diagram

[7.](#) YANG Module

This module imports types from [[RFC8819](#)],[[RFC8341](#)].

```
<CODE BEGINS> file "ietf-data-node-tags@2022-02-04.yang"
module ietf-data-node-tags {
  yang-version 1.1;
```

```

namespace "urn:ietf:params:xml:ns:yang:ietf-data-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 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 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

Wu, et al.

Expires 31 October 2022

[Page 11]

Internet-Draft

YANG Node Tags

April 2022

```
// and RFC number and remove this note.
revision 2022-02-04 {
  description
    "Initial revision.";
  reference
    "RFC XXXX: data node Tags in YANG Modules";
}
identity other-data-property {
  description
    "Base identity for data property type.";
}
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 nodes into the three
    categories, object, property, and metric. A data node
    tagged with 'object' tag can be one of container, leaf-list, or
    list. A data node tagged is with the 'property' tag is a leaf
    node. The data node tagged with the 'metric' tag can be one of
    container, leaf-list, list, or leaf. A data nodes tagged
    with either property tag or metric tag are child nodes
    belonging to a specific root data node. Each data node may
```

```

        contain one single 'object' tag, or one single 'property' tag,
        or one single 'metric' tag (these tags are mutually
        exclusive). As such, the origin of the value for the
        pre-defined tags should be set to 'system.'";
    }

    extension metric-type {
        argument tag;
        description
            "The argument 'tag' is of type 'tag'. The metric type can be
            used to provide metric data node classification
            (e.g., loss, jitter, packet loss, counter, gauge,
            summary, unknown) within a YANG module. The initial values of
            the 'metric-type' tag is defined in section 9.2, additional
            metric-type tag value can be added in the future.";
    }

    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 a metric

```

subobject.

The '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 an aggregated Ethernet interface) to be combined into aggregated statistics and reported as one metric subobject value.

The 'non-aggregated' multi-source aggregation type allows measurement from each source of the same type (e.g., line card, each subinterface of an aggregated Ethernet interface) to be reported separately.";

```

    }

    augment "/tags:module-tags/tags:module" {
        description
            "Augment the Module Tags module with data node tag

```

```

    attributes.";
container data-node-tags {
  description
    "Contains the list of data nodes and their associated data
    object tags.";
  list data-node {
    key "ni-id";
    description
      "Includes a list of data nodes and their associated data
      object tags.";
    leaf ni-id {
      type nacm:node-instance-identifier;
      mandatory true;
      description
        "The YANG data node name.";
    }
    leaf-list tag {
      type tags:tag;
      description
        "Lists the tags associated with the data node within
        the YANG module.

```

See the IANA 'YANG data node Tag Prefixes' registry for reserved prefixes and the IANA 'IETF YANG Data Object 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: data node Tags in YANG Data
  Modules, Section 9";
}
leaf-list masked-tag {
  type tags:tag;
  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
        object within YANG module, but they have no operational
        effect.";
    }
    leaf extended-tag-type {
        type identityref {
            base other-data-property;
        }
        description
        "Type of the extended tag. The extended tag type doesn't include opm t
        metric-type tag and multi-source tag three types defined in
        this document. The specific extended tag type and associated auxiliar
        are defined in the data node tags extension module.";
    }
}
}
}
}
<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 data node tag extension statements, a set of data node tags that are to be automatically associated with data node 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";
    }
    leaf bar {
      ntags:opm-tag "ietf:metric";
    }
  }
}
// ...
}

```

Figure 4: An Example of Data Object Tag

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

[9.](#) IANA Considerations

[9.1.](#) YANG Data Object Tag Prefixes Registry

This document requests IANA to create "YANG data node Tag Prefixes" subregistry in "YANG data 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 data 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 5: 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 Object Tags Registry

This document requests IANA to create "IETF OPM Tags", "IETF Metric Type Tags", "IETF Multiple Source Tags" three subregistries in "YANG data node Tag" registry. These 3 subregistries appear below "YANG data node Tag Prefixes" registry.

Three subregistries 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 subregistries 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 these three subregistries are as follows:

OPM Tag	Description	Reference
ietf:object	Represents Root object containing other data objects (e.g., interfaces)	[This document]
ietf:property	Represents a property data node (e.g., ifindex)	[This document]

Internet-Draft

YANG Node Tags

April 2022

	root object (e.g., interfaces)	
ietf:metric	Represent metric data object(e.g., ifstatistics) associated with specific root object(e.g., interfaces)	[This document]
+-----+-----+-----+		
Metric Type Tag	Description	Reference
+-----+-----+-----+		
ietf:delay	Represents the delay metric group to which the metric data nodes belong to.	[This document]
ietf:jitter	Represents the jitter metric group to which the metric data nodes belong to.	[This document]
ietf:loss	Represents the loss metric group to which the metric data nodes belong to.	[This document]
ietf:counter	Represents any metric value associated with a metric data node that monotonically increases over time, starting from zero.	[This document]
ietf:gauge	Represents current measurements associated with a metric data node that may increase, decrease or stay constant.	[This document]
ietf:summary	Represents the metric value associated with a metric data node that measures distributions of discrete	[This document]

	events without knowing predefined range.	
ietf:unknown	Represents the metric value associated with metric data node that can not determine the type of metric.	[This document]

Multiple Source Tag	Description	Reference
ietf:agg	Relates to multiple sources aggregation type (i.e., aggregated statistics)	[This document]
ietf:non-agg	Relates to multiple sources aggregation type (i.e., non-aggregated statistics)	[This document]

Figure 6: Table 2

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-data-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-data-node-tags
 namespace: urn:ietf:params:xml:ns:yang:ietf-data-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 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 data 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 data node tag meta-data may define various actions to be taken based on the data 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>>.

Wu, et al.

Expires 31 October 2022

[Page 19]

Internet-Draft

YANG Node Tags

April 2022

- [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>>.

Wu, et al.

Expires 31 October 2022

[Page 20]

Internet-Draft

YANG Node Tags

April 2022

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

[Appendix B](#). Instance Level Tunnel Tagging Example

In the example shown in Figure 2, 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>
    <data-node-tags xmlns="urn:ietf:params:xml:ns:yang:ietf-data-node-tags">
    <data-node>
    <ni-id>
    /tp:tunnel-svc[name='foo']/tp:packet-loss
    </ni-id>
    <tag>user:customer1_example_com</tag>
    <tag>ietf:critical</tag>
    </data-node>
    <data-node>
    <ni-id>
    /tp:tunnel-svc[name='bar']/tp:modified-time
    </ni-id>
    <tag>user:customer2_example_com</tag>
    </data-node>
    </data-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 Metric Type Tags" subregistry in [section 9.2](#).

[Appendix C](#). NETCONF Example

The following is a NETCONF example result from a query of the data 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:data-node-tags xmlns:s="urn:ietf:params:xml:ns:yang:ietf-\
data-node-tags">
        <s:data-node>
          <s:ni-id>/if:interfaces/if:interface</s:ni-id>
          <s:tag>ietf:object</s:tag>
        </s:data-node>
        <s:data-node>
          <s:ni-id>/if:interfaces/if:interface/if:last-change</\
s:ni-id>
          <s:tag>ietf:property</s:tag>
        </s:data-node>
        <s:data-node>
          <s:ni-id>
            /if:interfaces/if:interface/if:statistics/if:in-errors
          </s:ni-id>
          <s:tag>ietf:metric</s:tag>
          <s:tag>ietf:loss</s:tag>
          <s:tag>ietf:non-agg</s:tag>
        </s:data-node>
      </s:data-node-tags>
    </t:module>
    <t:module>
      <t:ni-id>ietf-ip</t:ni-id>
      <s:data-node-tags xmlns:s="urn:ietf:params:xml:ns:yang:ietf\
-data-node-tags">
        <s:data-node>
          <s:ni-id>/if:interfaces/if:interface/ip:ipv4</s:ni-id>
          <s:tag>ietf:object</s:tag>
        </s:data-node>
```

```

<s:data-node>
  <s:ni-id>/if:interfaces/if:interface/ip:ipv4/ip:enable\
</s:ni-id>
  <s:tag>ietf:property</s:tag>
</s:data-node>
<s:data-node>

```

```

  <s:ni-id>/if:interfaces/if:interface/ip:ipv4/ip:mtu</s:ni-id>
  <s:tag>ietf:metric</s:tag>
  <s:tag>ietf:non-agg</s:tag>
</s:data-node>
</s:data-node-tags>
</t:module>
</t:module-tags>
</ns0:data>

```

Figure 7: 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-data-node-tags-state@2022-02-03.yang"
module ietf-data-node-tags-state {
  yang-version 1.1;
  namespace
    "urn:ietf:params:xml:ns:yang:ietf-data-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

```

Wu, et al.

Expires 31 October 2022

[Page 25]

Internet-Draft

YANG Node Tags

April 2022

```

    <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: Data node Tags in YANG Data
      Modules";
}
identity other-data-property {
  description
    "Base identity for data property type.";
}
augment "/tags-s:module-tags-state/tags-s:module" {
  description
    "Augments the Module Tags module with data node tag
      attributes.";
```

```
container data-node-tags {
  config false;
  status deprecated;
  description
    "Contains the list of data nodes and their
      associated self describing tags.";
  list data-node {
    key "ni-id";
    status deprecated;
    description
      "Lists the data nodes and their associated self
        describing tags.";
    leaf ni-id {
      type nacm:node-instance-identifier;
      mandatory true;
      status deprecated;
      description
        "The YANG data node name.";
    }
    leaf-list tag {
      type tags:tag;
```

```

status deprecated;
description
  "Tags associated with the data node within the
  YANG module. See the IANA 'YANG data 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: Data node Tags in YANG Data
  Modules, Section 9";
}
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.";
  }
  leaf extended-tag-type {
    type identityref {
      base other-data-property;
    }
  }
  description
    "Type of the extended tag. The extended tag type doesn't
    include opm tag, metric-type tag and multi-source tag three
    types defined in this document. The specific extended tag
    type and associated auxiliary data are defined in the data

```

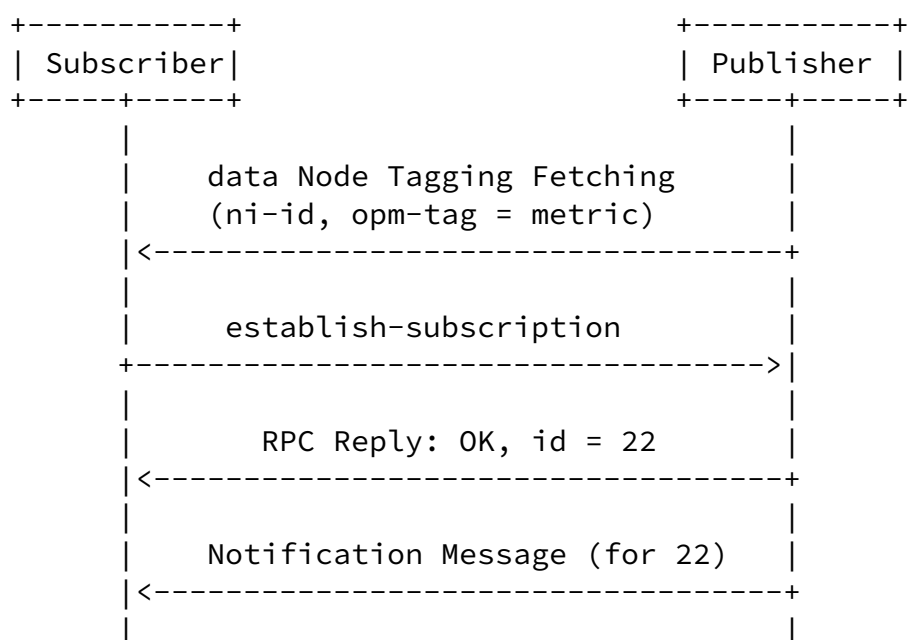
```

        node tags extension module.";
    }
}
}
}
}
<CODE ENDS>

```

Appendix E. Targeted Data Fetching Example

The following provides targeted data node collection example which helps reduce amount of data to be fetched. The subscription "id" values of 22 used below is just an example. In production, the actual values of "id" might not be small integers.



The subscriber fetches data node tag information from the provider using 'get-schema' operation. The data node tag information instruct the receiver to subscribe tagged data node (e.g., performance metric data nodes) using standard subscribed notification mechanism [[RFC8639](#)].

Figure 8 illustrates the retrieval of the list of available target

data nodes using the YANG instance file format [[RFC9195](#)]:

===== NOTE: '\\' line wrapping per [RFC 8792](#) =====

```
<?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>
    <t:module-tags xmlns:t="urn:ietf:params:xml:ns:yang:ietf-\
module-tags">
      <t:module>
        <t:name>ietf-interfaces</t:name>
        <s:data-node-tags xmlns:s="urn:ietf:params:xml:ns:yang:ietf-\
data-node-tags">
          <s:data-node>
            <s:ni-id>/if:interfaces/if:interface/if:in-errors</s:ni-id>
            <s:opm-tag>ietf:metric</s:opm-tag>
            <s:metric-type>ietf:loss</s:metric-type>
          </s:data-node>
        </s:data-node-tags>
      </t:module>
    </content-data>
  </instance-data-set>
```

Figure 8: List of Available Target Objects

With data node tag information carried in the <get-schema> operation, the subscriber identifies targeted 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. Alternatively, the subscriber can query data node tag list from somewhere (e.g., the network device, or offline document) using ietf-data-node-tags module defined in this document and fetch tagged data nodes 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.

===== 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>
```

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

[Appendix F](#). Changes between Revisions

Editorial Note (To be removed by RFC Editor)

v06 - v07

- * Update use case in [section 3](#) to remove object and subobject concept and massive related words.
- * Change the title into Data Node Tags in YANG Modules.

Internet-Draft

YANG Node Tags

April 2022

- * 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.
- * 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 data node Tag registry and YANG data 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

Wu, et al.

Expires 31 October 2022

[Page 31]

Internet-Draft

YANG Node Tags

April 2022

- * 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 data 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 data 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

Wu, et al.

Expires 31 October 2022

[Page 32]

Internet-Draft

YANG Node Tags

April 2022

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

