NETMOD Working Group
Internet-Draft
Intended status: Standards Track
Expires: February 17, 2021

Q. Wu
Huawei
B. Claise
Cisco
L. Geng
Z. Du
China Mobile

August 16, 2020

Self Explanation Data Object Tags
draft-tao-netmod-yang-node-tags-05

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 identify characteristics data and correlate data objects from different data sources and 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.

An extension statement to be used to indicate self explanation YANG data object tags that SHOULD be added by the module implementation automatically (i.e., outside of configuration).

A YANG module [RFC7950] is defined, which augments Module tag model and provides a list of data object entries to allow for adding or removing of self explanation data object tags as well as viewing the set of self explanation tags associated with the data object within the YANG module.

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 February 17, 2021.

Table of Contents

1. Introduction 3
1.1. Self Explanation Data Object Tags Use Cases 4
1.1.1. Network Performance Data Collection 4
1.1.2. Context Information Tagging 7
1.2. Terminology 8
2. Data Object Tag Values 8
2.1. IETF Tags Prefix 8
2.2. Vendor Tags Prefix 9
2.3. User Tags Prefix 9
2.4. Reserved Tags Prefix 9
3. Data Object Tag Management 9
3.1. Module Design Tagging 9
3.2. Implementation Tagging 10
3.3. User Tagging 11
4. Tags Module Structure 11
4.1. Tags Module Tree 11
5. YANG Module 11
6. Guidelines to Model Writers 19
6.1. Define Standard Tags 19
7. IANA Considerations 20
7.1. YANG Data Object Tag Prefixes Registry 20
7.2. IETF YANG Data Object Tags Registry 20
7.3. Updates to the IETF XML Registry 25
7.4. Updates to the YANG Module Names Registry 26
8. Security Considerations 26
9. Contributors 26

10. References 27
10.1. Normative References 27
10.2. Informative References 27
Appendix A. NETCONF Example 28
Appendix B. Non-NMDA State Module 29
Appendix C. Targeted data object subscription example 37
Authors' Addresses 40

1. Introduction

As described [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 Souce 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 (e.g., performance metric specific data object type) at different locations in various different ways, lack consistent classification criteria and representation granularity,e.g., sensor data in hardware model is defined with fine granularity with value scale and value precision while interface model only provides statistics data for specific interface type.

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

- Provide dictionary meaning for specific targeted data objects;
- Indicate relationship between data objects within the same YANG module or from different YANG modules;
- Identify key performance metric scale, precision, statistics operation type;
- Identify specific service or feature, data source.

The self explanation data object tags can be used by the client to identify characteristics data and correlate data objects from different data sources and 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 explanation data object tags supported by a NETCONF server via <get-schema> operation. The self explanation 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 self explanation 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 explanation tags as well as viewing the set of self explanation tags associated with a data object within YANG modules.

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

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

1.1. Self Explanation Data Object Tags Use Cases

The following is a list of already implemented and potential use cases.

1.1.1. Network Performance Data Collection

Among Data object tags, the performance metric tags can be used to capture the performance metrics and properties associated with YANG data nodes or data objects modelled with YANG (See Figure 1).

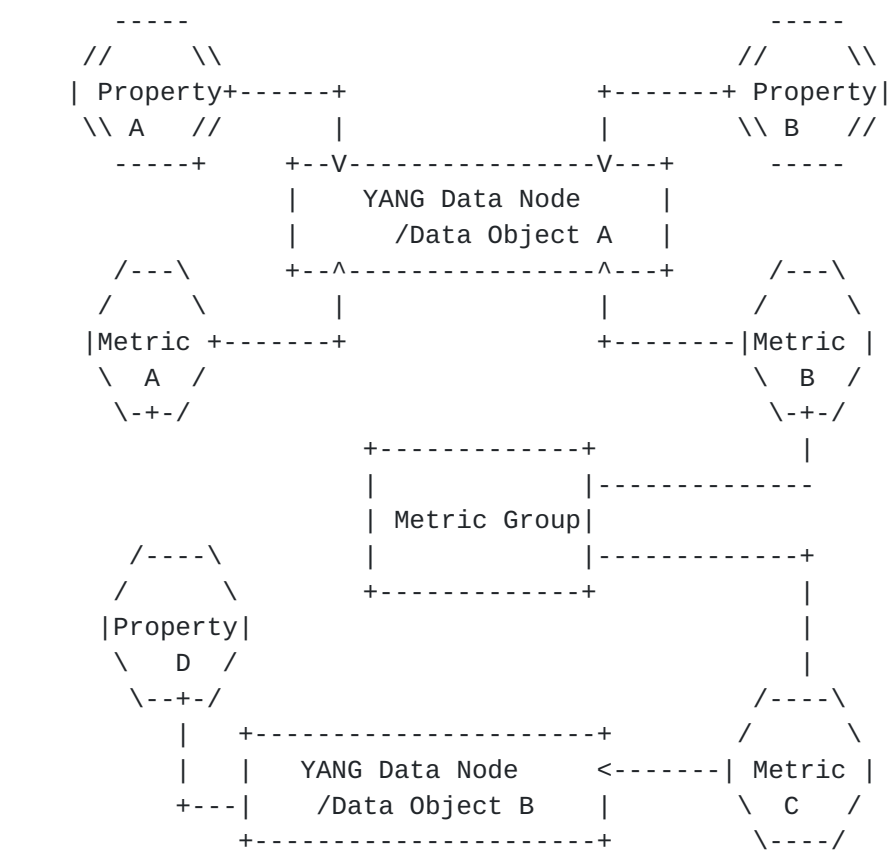


Figure 1

The use of performance metric tags would be to help filter discrete categories of YANG data objects across different YANG modules supported by a device and capture network performance data. If data objects across YANG modules are suitably tagged and learnt by the client from a live server, the client can extract paths to all interested data objects and then use an XPath query to list all related data objects which reflect network characteristics(see Figure 2).

Object Name	Property Name	Metric Group	Metric Name	Module
tunnel-svc	name	-	-	tunnel
tunnel-svc	create-time	-	-	tunnel
tunnel-svc	modified-time	-	-	
tunnel-svc	-	lsp-ping-pm	avg-latency	tunnel-pm
tunnel-svc	-	lsp-ping-pm	packet-loss	tunnel-pm
tunnel-svc	-	lsp-ping-pm	min-latency	tunnel-pm
tunnel-svc	-	lsp-ping-pm	max-latency	tunnel-pm
tunnel-svc	-	lsp-ping-pm	transmitted -packet	tunnel-pm

Metric Group	Metric Name	Metric Precision	Metric Scale	Operation Type
lsp-ping-pm	avg-latency	1	1	avg
lsp-ping-pm	packet-loss	1	1	avg
lsp-ping-pm	min-latency	1	1	min
lsp-ping-pm	max-latency	1	1	max
lsp-ping-pm	transmitted -packet	1	1	normal

Figure 2

1.1.2. Context Information Tagging

Performance metric tags can also be used to help correlate data objects with the same characteristics when clients are interacting with various different devices with the different categories of YANG data node across different YANG modules. For example, one management client could mark some specific data node across modules implemented in various different devices with the same metric group tag as context information, so consistent representation and reporting can be provided for YANG data nodes belonging to the same metric group (see Figure 2).

Another example is the management client could mark some data node across different level of YANG modules implemented in the device, the management system separately with the same service tag (e.g., L3VPN Service) as context information, so root cause can be identified efficiently during network failure troubleshooting (See Figure 3)

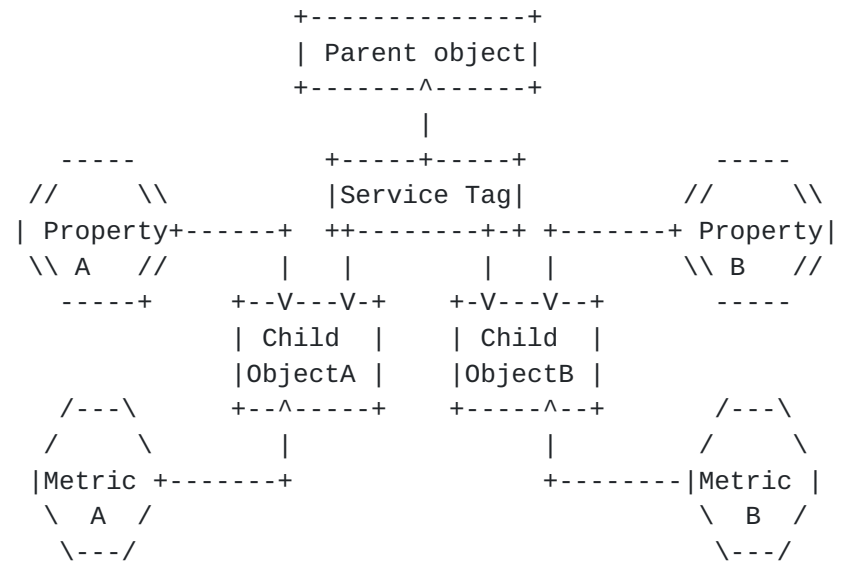


Figure 3

Service Tag	Metric Group	Metric Name	Module	Level
L3VPN	L3VPN	maximum-routes	L3VPN	Service
L3VPN	OSPF-Process	total-active routes	OSPF	Device
L3VPN	RIP-Process	total-active routes	RIP	Device
L3VPN	BGP-Process	total-active routes	BGP	Device

Figure 4

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 node 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 type, property and metric three categories.

metric-scale extension statement: Provide an additional metric scale (e.g., Measurement scaling factor of 10^0, 10^-3,10^3) information associated with the performance metric data object within YANG module. ;

metric-precision extension statement: Provide an additional metric precision (e.g., the range -8 to -1, 0, the range 1 to 9)

information associated with the performance metric data object within YANG module. ;

operation-type extension statement: Statistics operation of performance metric data object within YANG module, The statistics operation include normal,min, max, average, sum,threshold.

metric-group extension statement: Provide correlation between different metric information associated with data object within YANG module.

data-source extension statement: The data source type (e.g., connectivity,resource, hardware ,qos,policy) associated with the performance metric data object within YANG module.

multi-source-tag extension statement: Identify multiple source aggregation tye(e.g., line card,member link in an aggregated Ethernet interface) related to performance metric data object or interface related data object). Two source aggregation source types are supported, one is aggregation, the other is membership.

service-tag extension statement: Provide a service classification information (e.g., tunnel, l3vpn,l2vpn) information associated with data object within YANG module;

task-tag extension statement: Provide a task classification information (e.g., fault management, performance measurement) information associated with data object within YANG module.

Among these extension statements, the opm-tag extension statement,meric-scale extension statement, metric-precision extension statement,operation-type extension statement are performance meric related while metric-group extension statement,data-source extension statement, multi-source-tag extension statement, service-tag extension statement, task-tag extension statement are context information related.

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 7.2, 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. Tags Module Structure

4.1. 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>";
```

```
  description
    "This module describes a mechanism associating self-explanation
     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";
  }

  typedef metric-precision {
    type int8 {
      range "-8 .. 9";
    }
    description
      "A node using this data type represents a sensor value
       precision range.

       A node of this type SHOULD be defined together with nodes of
       type measurement-units and type measurement-scale.  Together,
       associated nodes of these three types are used to identify the
       semantics of a node of type sensor-value.

       If a node of this type contains a value in the range 1 to 9,
       it represents the number of decimal places in the fractional
       part of an associated sensor-value fixed-point number.
       If a node of this type contains a value in the range -8 to -1,
       it represents the number of accurate digits in the associated
       sensor-value fixed-point number.

       The value zero indicates the associated sensor-value node is
       not a fixed-point number.
```

```
       Server implementers must choose a value for the associated
       sensor-value-precision node so that the precision and accuracy
       of the associated sensor-value node is correctly indicated.

       For example, a component representing a temperature sensor
       that can measure 0 to 100 degrees C in 0.1 degree
       increments, +/- 0.05 degrees, would have a
       sensor-value-precision value of '1', a sensor-value-scale
       value of 'units', and a sensor-value ranging from '0' to
       '1000'.  The sensor-value would be interpreted as
       'degrees C * 10'.";
    reference
      "RFC 3433: Entity Sensor Management Information Base -
                 EntitySensorPrecision";
  }

  typedef metric-scale {
    type enumeration {
      enum yocto {
        value 1;
        description
          "Measurement scaling factor of 10^-24.";
      }
      enum zepto {
        value 2;
        description
          "Measurement scaling factor of 10^-21.";
      }
      enum atto {
        value 3;
        description
          "Measurement scaling factor of 10^-18.";
      }
      enum femto {
        value 4;
        description
          "Measurement scaling factor of 10^-15.";
      }
      enum pico {
        value 5;
        description
          "Measurement scaling factor of 10^-12.";
      }
      enum nano {
        value 6;
        description
          "Measurement scaling factor of 10^-9.";
      }
```

```
      enum micro {
        value 7;
        description
          "Measurement scaling factor of 10^-6.";
      }
      enum milli {
        value 8;
        description
          "Measurement scaling factor of 10^-3.";
      }
      enum units {
        value 9;
        description
          "Measurement scaling factor of 10^0.";
      }
      enum kilo {
        value 10;
        description
          "Measurement scaling factor of 10^3.";
      }
      enum mega {
        value 11;
        description
          "Measurement scaling factor of 10^6.";
      }
      enum giga {
        value 12;
        description
          "Measurement scaling factor of 10^9.";
      }
      enum tera {
        value 13;
        description
          "Measurement scaling factor of 10^12.";
      }
      enum peta {
        value 14;
        description
          "Measurement scaling factor of 10^15.";
      }
      enum exa {
        value 15;
        description
          "Measurement scaling factor of 10^18.";
      }
      enum zetta {
        value 16;
        description
```

```
          "Measurement scaling factor of 10^21.";
      }
      enum yotta {
        value 17;
        description
          "Measurement scaling factor of 10^24.";
      }
    }
    description
      "A node using this data type represents a data scaling factor,
       represented with an International System of Units (SI) prefix.
       The actual data units are determined by examining a node of
       this type together with the associated sensor-value-type.

       A node of this type SHOULD be defined together with nodes of
       type sensor-value-type and type sensor-value-precision.
       Together, associated nodes of these three types are used to
       identify the semantics of a node of type sensor-value.";
    reference
      "RFC 3433: Entity Sensor Management Information Base -
                 EntitySensorDataScale";
  }

  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 type, property, and metric
       As such the origin of the value for the pre-defined tags should be
       set to 'system'[RFC8342].";
  }

  extension metric-scale {
    argument tag;
    description
      "The argument 'tag' is of type 'tag'. The metric-scale tag can be
       used to provide an additional data scale factor(e.g., Measurement
       scaling factor of 10^0, 10^-3,10^3) information associated with
       the performance metric data object.

       A node using metric scale tag SHOULD be defined together with nodes of
       type metric unit and type metric precision.
       Together, associated nodes of these three types are used to
       identify the semantics of the performance metric data object.";
    reference
      "RFC 3433: Entity Sensor Management Information Base -
```

```
                EntitySensorDataScale";
  }

  extension metric-precision {
    argument tag;
    description
      "The argument 'tag' is of type 'tag'. The metric-precision can be
       used to provide an additional sensor value precision range (e.g.,
       the range -8 to -1, 0, the range 1 to 9) information associated
       with the performance metric data object.

       A node using metric precision tag SHOULD be defined together with
       nodes of type metric unit and type metric scale. Together, associated
       nodes of these three types are used to identify the semantics of the
       performance metric data object.

       If a node of this type contains a value in the range 1 to 9,
       it represents the number of decimal places in the fractional
       part of an associated sensor-value fixed-point number.
       If a node of this type contains a value in the range -8 to -1,
       it represents the number of accurate digits in the associated
       sensor-value fixed-point number.

       The value zero indicates the associated sensor-value node is
       not a fixed-point number.

       Server implementers must choose a value for the associated
       metric precision tag so that the precision and accuracy
       of the associated sensor-value node is correctly indicated.

       For example, a component representing a temperature sensor
       that can measure 0 to 100 degrees C in 0.1 degree
       increments, +/- 0.05 degrees, would have a
       sensor-value-precision value of '1', a sensor-value-scale
       value of 'units', and a sensor-value ranging from '0' to
       '1000'.  The sensor-value would be interpreted as
       'degrees C * 10'.";
    reference
      "RFC 3433: Entity Sensor Management Information Base -
                 EntitySensorDataScale";
  }

  extension operation-type {
    argument tag;
    description
      "The argument 'tag' is of type 'tag'.The statistics-operation can be
       used to provide an additional statistics operation type(e.g., sum,
       min, max,sum,last, threshold) information associated with the
```

performance metric

```
       data object.

       If the operation type is threshold type, the corresponding
       data object support threshold handling,e.g.,scan all interfaces
       for a certain type every 5 seconds and check the counters or
       status to cross threshold, return an array of interface entries
       that match the search.

       If the operation type is average,min,max,sum,last,
       it indicate the data object supports statistics operation, e.g.,
       scan all interfaces for a certain type every 5 seconds up to 60 seconds,
       only return min, average, max, sum value of specific data object rather
than
       the values that are current at the end of 60 seconds.";
  }

  extension metric-group {
    argument tag;
    description
      "The argument 'tag' is of type 'tag'.The metric-group can be
       used to provide correlation between different metric information
       associated with YANG data object.";
  }

  extension service-tag {
    argument tag;
    description
      "The argument 'tag' is of type 'tag'.The service-tag can be
       used to provide a service classification information (e.g., tunnel,
       l3vpn,l2vpn) information associated with YANG data object.";
  }

  extension task-tag {
    argument tag;
    description
      "The argument 'tag' is of type 'tag'.The task-tag can be
       used to provide a task classification information (e.g., fault
management,
       performance measurement) information associated with YANG data object.";
  }

  extension data-source {
    argument tag;
    description
      "The argument 'tag' is of type 'tag'.The data-source-type can be
       used to provide an additional data source type (e.g., connectivity,
       resource, hardware,qos,policy) information associated with
       the performance metric data object.";
```

```
}
```

```
  extension multi-source-tag {
    argument tag;
    description
      "The argument 'tag' is of type 'tag'.The multi-source-tag can be
       used to identify multiple source aggregation tye(e.g., line card,
       member link in an aggregated Ethernet interface) related to performance
       metric data object or interface related to data object).

       Two source aggregation source types are supported, one is aggregation
       which groups data from two or multiple different data objects,
       the other is membership which identify each data object(e.g.,
       linecard, member link from multiple source aggregation.";
  }

  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 self
explanation tags";
      list data-object {
        key "object-name";
        description
          "A list of self explanation nodes and their associated 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 YANG module. See
             the IANA 'YANG Data Object Tag Prefixes' registry for reserved
             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 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 node-tag extension statements, a set of tags that are to be automatically associated with it (i.e., not added through configuration).

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

The module writer can use existing standard tags, or use new tags

defined in the model 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	[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 nine new registries "IETF Data Object Tags","IETF Metric Precision Tags","IETF Metric Scale Tags","IETF Operation Type Tags","IETF Metric Group Tags","IETF Data Source Tags","IETF Multiple Source Tags","IETF Service Tags","IETF Task Tags" grouped under a new "Protocol" category. These nine registries

should be included below "YANG Data Object Tag Prefixes" when listed on the same page.

Nine 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 four registries is IETF Review [RFC8126].

The initial values for these nine registries are as follows.

OPM Tag	Description	Reference
ietf:object-type	Relates to object type (e.g., interfaces).	[This document]
ietf:metric	Relates to performance metric info (e.g., ifstatistics).	[This document]
ietf:property	Represents a object property (e.g.,ifindex).	[This document]

Metric Precision Tag	Description	Reference
ietf:minus-eight	Relates to metric precision of performance metric	[This document]
ietf:minus-seven	Relates to metric precision of performance metric	[This document]
ietf:minus-six	Relates to metric precision of performance metric	[This document]
ietf:minus-five	Relates to metric precision of performance metric	[This document]
ietf:minus-four	Relates to metric precision of performance metric	[This document]
ietf:minus-three	Relates to metric precision of performance metric	[This document]

ietf:minus-two	Relates to metric precision of performance metric	[This document]
ietf:minus-one	Relates to metric precision of performance metric	[This document]
ietf:zero	Relates to metric precision of performance metric	[This document]
ietf:one	Relates to metric precision of performance metric	[This document]
ietf:two	Relates to metric precision of performance metric	[This document]
ietf:three	Relates to metric precision of performance metric	[This document]
ietf:four	Relates to metric precision of performance metric	[This document]
ietf:five	Relates to metric precision of performance metric	[This document]
ietf:six	Relates to metric precision of performance metric	[This document]
ietf:seven	Relates to metric precision of performance metric	[This document]
ietf:eight	Relates to metric precision of performance metric	[This document]
ietf:nine	Relates to metric precision of performance metric	[This document]

Metric Scale Tag	Description	Reference
ietf:yocto	Relates to metric scale of performance metric	[This document]
ietf:zepto	Relates to metric scale of performance metric	[This document]
ietf:atto	Relates to metric scale of performance metric	[This document]

ietf: femto	Relates to metric scale of performance metric	[This document]
ietf: pico	Relates to metric scale of performance metric	[This document]
ietf: nano	Relates to metric scale of performance metric	[This document]
ietf: micro	Relates to metric scale of performance metric	[This document]
ietf: milli	Relates to metric scale of performance metric	[This document]
ietf: units	Relates to metric scale of performance metric	[This document]
ietf: kilo	Relates to metric scale of performance metric	[This document]
ietf: mega	Relates to metric scale of performance metric	[This document]
ietf: giga	Relates to metric scale of performance metric	[This document]
ietf: tera	Relates to metric scale of performance metric	[This document]
ietf: peta	Relates to metric scale of performance metric	[This document]
ietf: exa	Relates to metric scale of performance metric	[This document]
ietf: zetta	Relates to metric scale of performance metric	[This document]
ietf: yotta	Relates to metric scale of performance metric	[This document]

Operation Type Tag	Description	Reference
ietf:normal	Relates to statistics operation(e.g.,average,	[This document]

	min, max, normal,etc)	
ietf:avg	Relates to statistics operation(e.g.,average, min, max, sum,etc)	[This document]
ietf:sum	Relates to statistics operation(e.g.,average, min, max, sum,etc)	[This document]
ietf:min	Relates to statistics operation(e.g.,average, min, max, sum,etc)	[This document]
ietf:max	Relates to statistics operation(e.g.,average, min, max, sum,etc)	[This document]
ietf:threshold	Relates to statistics operation(e.g.,average, min, max, threshold,etc)	[This document]

Metric Group Tag	Description	Reference
ietf:delay	Represent metric group (e.g., loss, jitter,delay)	[This document]
ietf:jitter	Represent metric group (e.g., loss, jitter,delay)	[This document]
ietf:loss	Represent metric group (e.g., loss, jitter,delay)	[This document]

Multiple Source Tag	Description	Reference
ietf:member	Relates to multiple source aggregation type(e.g., lag, linecard, sub inf)	[This document]
ietf:agg	Relates to multiple source aggregation type(e.g.,agg)	[This document]

Data Source Tag	Description	Reference
ietf:service-flow	Relates to data source type(e.g., microburst).	[This document]
ietf:topo	Relates to data source type(e.g., topology).	[This document]

ietf:resource	Relates to data source type info (e.g., interface,queue).	[This document]
ietf:policy	Relates to data source type info (e.g., acl, routing policy	[This document]
ietf:hardware	Relates to data source type (e.g.,optical module).	[This document]

Service Tag	Description	Reference
ietf:l3vpn	Relates to service offering(e.g.,l3vpn l2vpn,tunnel,etc)	[This document]
ietf:l2vpn	Relates to service offering(e.g.,l3vpn l2vpn,tunnel,etc)	[This document]
ietf:te-tunnel	Relates to service offering(e.g.,l3vpn l2vpn,tunnel,etc)	[This document]

Task Tag	Description	Reference
ietf:vpn-diag	Relates to vpn serivce diagonostic function	[This document]
ietf:vpn-fullfilment	Relates to vpn service fullfillment function	[This document]
ietf:vpn-assurance	Relates to vpn service assurance function	[This document]

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. Contributors

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

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>.

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 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/if:statistics/if:in-errors
         </s:object-name>
         <s:tag>ietf:metric</s:tag>
         <s:tag>ietf:avg</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:type</s:object-name>
        <s:tag>ietf:object-type</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/ip:mtu</s:object-
name>
        <s:tag>ietf:metric</s:tag>
        <s:tag>ietf:normal</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</s:object-name>
         <s:tag>ietf:object-type</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 {
  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>";
  description
    "This module describes a mechanism associating self-explanation
     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";
  }
```

```
  typedef metric-precision {
    type int8 {
      range "-8 .. 9";
    }
    description
      "A node using this data type represents a sensor value
       precision range.

       A node of this type SHOULD be defined together with nodes of
       type measurement-units and type measurement-scale.  Together,
       associated nodes of these three types are used to identify the
       semantics of a node of type sensor-value.

       If a node of this type contains a value in the range 1 to 9,
       it represents the number of decimal places in the fractional
       part of an associated sensor-value fixed-point number.
       If a node of this type contains a value in the range -8 to -1,
       it represents the number of accurate digits in the associated
       sensor-value fixed-point number.

       The value zero indicates the associated sensor-value node is
       not a fixed-point number.

       Server implementers must choose a value for the associated
       sensor-value-precision node so that the precision and accuracy
       of the associated sensor-value node is correctly indicated.

       For example, a component representing a temperature sensor
       that can measure 0 to 100 degrees C in 0.1 degree
       increments, +/- 0.05 degrees, would have a
       sensor-value-precision value of '1', a sensor-value-scale
       value of 'units', and a sensor-value ranging from '0' to
       '1000'.  The sensor-value would be interpreted as
       'degrees C * 10'.";
    reference
      "RFC 3433: Entity Sensor Management Information Base -
                 EntitySensorPrecision";
  }

  typedef metric-scale {
    type enumeration {
      enum yocto {
        value 1;
        description
          "Measurement scaling factor of 10^-24.";
      }
      enum zepto {
        value 2;
```

```
        description
          "Measurement scaling factor of 10^-21.";
      }
      enum atto {
        value 3;
        description
          "Measurement scaling factor of 10^-18.";
      }
      enum femto {
        value 4;
        description
          "Measurement scaling factor of 10^-15.";
      }
      enum pico {
        value 5;
        description
          "Measurement scaling factor of 10^-12.";
      }
      enum nano {
        value 6;
        description
          "Measurement scaling factor of 10^-9.";
      }
      enum micro {
        value 7;
        description
          "Measurement scaling factor of 10^-6.";
      }
      enum milli {
        value 8;
        description
          "Measurement scaling factor of 10^-3.";
      }
      enum units {
        value 9;
        description
          "Measurement scaling factor of 10^0.";
      }
      enum kilo {
        value 10;
        description
          "Measurement scaling factor of 10^3.";
      }
      enum mega {
        value 11;
        description
          "Measurement scaling factor of 10^6.";
      }
```

```
      enum giga {
        value 12;
        description
          "Measurement scaling factor of 10^9.";
      }
      enum tera {
        value 13;
        description
          "Measurement scaling factor of 10^12.";
      }
      enum peta {
        value 14;
        description
          "Measurement scaling factor of 10^15.";
      }
      enum exa {
        value 15;
        description
          "Measurement scaling factor of 10^18.";
      }
      enum zetta {
        value 16;
        description
          "Measurement scaling factor of 10^21.";
      }
      enum yotta {
        value 17;
        description
          "Measurement scaling factor of 10^24.";
      }
    }
    description
      "A node using this data type represents a data scaling factor,
       represented with an International System of Units (SI) prefix.
       The actual data units are determined by examining a node of
       this type together with the associated sensor-value-type.

       A node of this type SHOULD be defined together with nodes of
       type sensor-value-type and type sensor-value-precision.
       Together, associated nodes of these three types are used to
       identify the semantics of a node of type sensor-value.";
    reference
      "RFC 3433: Entity Sensor Management Information Base -
                 EntitySensorDataScale";
  }

  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 type, property, and
       metric. As such the origin of the value for the pre-defined tags
       should be set to 'system'[RFC8342].";
  }

  extension metric-scale {
    argument tag;
    description
      "The argument 'tag' is of type 'tag'. The metric-scale tag can be
       used to provide an additional data scale factor(e.g., Measurement
       scaling factor of 10^0, 10^-3,10^3) information associated with
       the performance metric data object.

       A node using metric scale tag SHOULD be defined together with nodes of
       type metric unit and type metric precision.
       Together, associated nodes of these three types are used to
       identify the semantics of the performance metric data object.";
    reference
      "RFC 3433: Entity Sensor Management Information Base -
                 EntitySensorDataScale";
  }

  extension metric-precision {
    argument tag;
    description
      "The argument 'tag' is of type 'tag'. The metric-precision can be
       used to provide an additional sensor value precision range (e.g.,
       the range -8 to -1, 0, the range 1 to 9) information associated
       with the performance metric data object.

       A node using metric precision tag SHOULD be defined together with
       nodes of type metric unit and type metric scale. Together, associated
       nodes of these three types are used to identify the semantics of the
       performance metric data object.

       If a node of this type contains a value in the range 1 to 9,
       it represents the number of decimal places in the fractional
       part of an associated sensor-value fixed-point number.
       If a node of this type contains a value in the range -8 to -1,
       it represents the number of accurate digits in the associated
       sensor-value fixed-point number.

       The value zero indicates the associated sensor-value node is
       not a fixed-point number.
```

```
       Server implementers must choose a value for the associated
       metric precision tag so that the precision and accuracy
       of the associated sensor-value node is correctly indicated.

       For example, a component representing a temperature sensor
       that can measure 0 to 100 degrees C in 0.1 degree
       increments, +/- 0.05 degrees, would have a
       sensor-value-precision value of '1', a sensor-value-scale
       value of 'units', and a sensor-value ranging from '0' to
       '1000'.  The sensor-value would be interpreted as
       'degrees C * 10'.";
    reference
      "RFC 3433: Entity Sensor Management Information Base -
                 EntitySensorDataScale";
  }

  extension operation-type {
    argument tag;
    description
      "The argument 'tag' is of type 'tag'.The statistics-operation can be
       used to provide an additional statistics operation type(e.g., sum,
       min, max,sum,last, threshold) information associated with the
performance metric
       data object.

       If the operation type is threshold type, the corresponding
       data object support threshold handling,e.g.,scan all interfaces
       for a certain type every 5 seconds and check the counters or
       status to cross threshold, return an array of interface entries
       that match the search.

       If the operation type is average,min,max,sum,last,
       it indicate the data object supports statistics operation, e.g.,
       scan all interfaces for a certain type every 5 seconds up to 60 seconds,
       only return min, average, max, sum value of specific data object rather
than
       the values that are current at the end of 60 seconds.";
  }

  extension service-tag {
    argument tag;
    description
      "The argument 'tag' is of type 'tag'.The service-tag can be
       used to provide a service classification information (e.g., tunnel,
       l3vpn,l2vpn) information associated with YANG data object.";
  }

  extension task-tag {
```

```
    argument tag;
    description
```

```
      "The argument 'tag' is of type 'tag'.The task-tag can be
       used to provide a task classification information (e.g., fault
management,
       performance measurement) information associated with YANG data object.";
  }

  extension data-source {
    argument tag;
    description
      "The argument 'tag' is of type 'tag'.The data-source-type can be
       used to provide an additional data source type (e.g., connectivity,
       resource, hardware,qos,policy) information associated with
       the performance metric data object tag.";
  }

  extension multi-source-tag {
    argument tag;
    description
      "The argument 'tag' is of type 'tag'.The multi-source-tag can be
       used to identify multiple source aggregation tye(e.g., line card,
       member link in an aggregated Ethernet interface) related to performance
       metric data objects or interface related data objects).

       Two source aggregation source types are supported, one is aggregation
       which groups data from two or multiple different data objects,
       the other is membership which identify each data object(e.g.,
       linecard, member link from multiple source aggregation.";
  }

  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
explanation tags";
      list data-object {
        key "object-name";
        status deprecated;
        description
          "A list of self explanation nodes and their associated tags";
        leaf object-name {
          type nacm:node-instance-identifier;
          mandatory true;
          status deprecated;
          description
```

```
            "The YANG data object name.";
        }
```

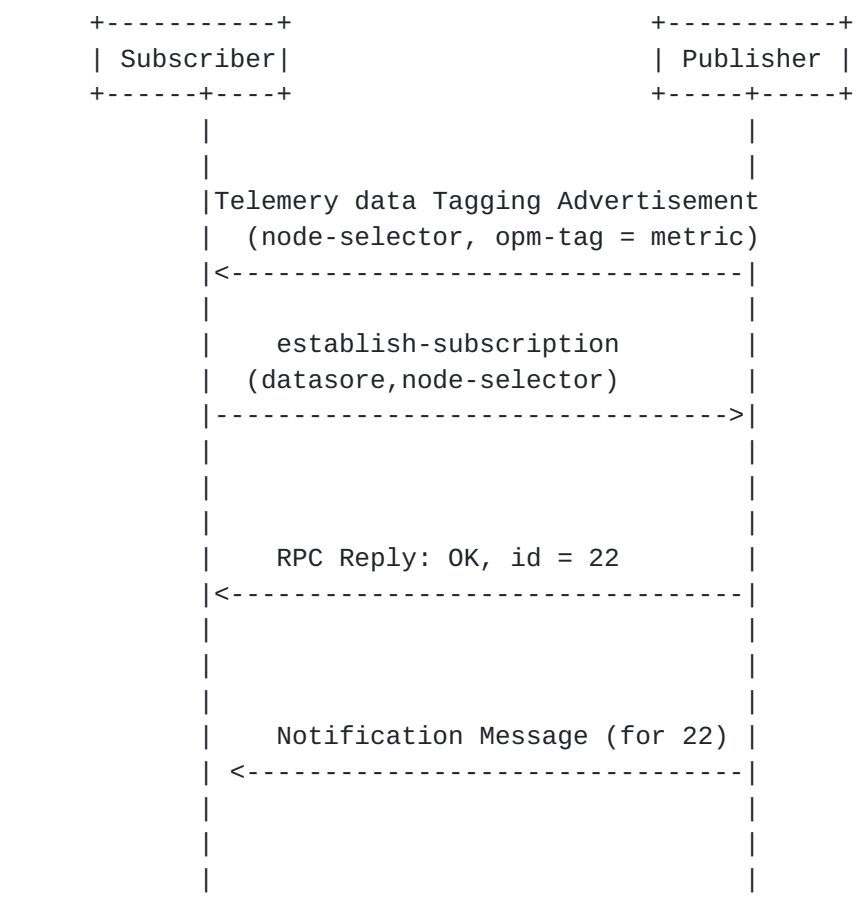
```
        leaf-list tag {
          type tags:tag;
          status deprecated;
          description
            "Tags associated with the data object within YANG module. See
             the IANA 'YANG Data Object Tag Prefixes' registry for reserved
             prefixes and the IANA'IETF YANG Data 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;
          status deprecated;
          description
            "The list of tags that should not be associated with the data
             object within 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>
```

Appendix C. Targeted data object subscription example

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



The publisher advertise telemetry data object capability to the subscriber to instruct the receiver to subscribe targeted data object with specific characteristics (e.g., performance metric data object) and specific data path corresponding to the targeted data object.

The following XML example [W3C.REC-xml-20081126] illustrates the advertisment of the list of available target objects:

```
<?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:self-tag-id>counter</sec:self-tag-id>
            <sec:opm-tag>metric</sec:opm-tag>
            <sec:operation-type>avg</sec:operation-type>
          </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 establish-subscription RPC to subscribe specific 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 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: gengliang@chinamobile.com

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

Email: duzongpeng@chinamobile.com