

OPSAWG
Internet-Draft
Intended status: Standards Track
Expires: 23 April 2026

B. Claise
Everything OPS
J. Quilbeuf
Huawei
D. Lopez
I. Dominguez
Telefonica I+D
T. Graf
Swisscom
20 October 2025

A Data Manifest for Contextualized Telemetry Data
draft-ietf-opsawg-collected-data-manifest-10

Abstract

Network platforms use Network Telemetry, such as YANG-Push, to continuously stream information, including both counters and state information. This document describes the metadata that ensure that the collected data can be interpreted correctly. This document specifies the Data Manifest, composed of two YANG data models (the Platform Manifest and the non-normative Data Collection Manifest). These YANG modules are specified at the network level (e.g., network controllers) to provide a model that encompasses several network platforms. The Data Manifest must be streamed and stored along with the data, up to the collection and analytics systems to keep the collected data fully exploitable by the data scientists and relevant tools. Additionally, this document specifies an augmentation of the YANG-Push model to include the actual collection period, in case it differs from the configured collection period.

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 23 April 2026.

Copyright Notice

Copyright (c) 2025 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include 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. Use Cases	5
3.1. Network Analytics	5
3.2. New Device Onboarding	6
3.3. Data Mesh Principles in Networking	6
4. The "ietf-yp-current-period" YANG module	7
5. Platform Manifest	9
5.1. Overview of the Model	9
5.2. "ietf-platform-manifest" YANG module	11
6. Data Collection Manifest	14
6.1. Overview of the Model	15
6.2. The "example-collection-manifest" YANG module	17
7. Data Manifest and the Collected Data	19
7.1. Collecting the Data Manifest	19
7.2. Mapping Collected Data to the Data Manifest	20
8. Operational Considerations	21
9. Example	22
10. Security Considerations	24
11. IANA Considerations	25
12. Contributors	26
13. Open Issues	26
14. Normative References	26
15. Informative References	27
Appendix A. Changes between revisions	30
Appendix B. An Example of Use Based on MDT	34
Appendix C. Generating YANG Tree Diagrams	37
Appendix D. Validating the Example	41
Acknowledgements	42
Authors' Addresses	42

1. Introduction

Network platforms use Network Telemetry, such as YANG-Push [RFC8641], to continuously stream information, including both counters and state information.

This document specifies what needs to be kept as metadata to ensure that the collected data can still be interpreted correctly throughout the collection and network analytics toolchain. When streaming YANG-structured data with YANG-Push, there is a semantic definition in the corresponding YANG module definition. This is the semantic information for the collected data nodes: While this semantic is absolutely required to correctly decode and interpret the data, understanding the network platform and collection environment contexts information is equally important to interpret the data.

One part of this information is the actual collection period, as opposed to the configured collection period. On some platforms, that period can be adjusted unilaterally by the platform, for instance to reduce the load incurred by sending the telemetry. To later exploit the collected data, getting this actual collection period is crucial. This document defines a YANG model augmenting the YANG-Push model [RFC8641] to expose the actual collection period in Section 4.

This document introduces the Data Manifest, which is composed of two YANG modules, namely, the Platform Manifest and the data collection manifest, to keep the collected data exploitable by the data scientists and relevant tools.

The Platform Manifest contains information characterizing the platform streaming the telemetry information, while the Data Collection Manifest contains the required information to characterize how and when the telemetry information was metered. The Platform Manifest is specified in Section 5. An example of Data Collection Manifest is specified in Section 6. The latter module is non-normative due to the lack of design-time schema mount in YANG, see Section 1 of [RFC8528].

These two YANG modules do not expose any new information but rather define what should be exposed by a platform streaming or storing telemetry data. Some related YANG modules have been specified to retrieve the platform capabilities such as:

- * "YANG Library" [RFC8525].
- * "YANG Modules Describing Capabilities for Systems and Datastore Update Notifications" [RFC9196] for the platform capabilities regarding the production and export of telemetry data.

- * [I-D.claise-netconf-metadata-for-collection], which is based on [RFC9196] to define the optimal settings to stream specific items (i.e., per path).

These related YANG modules are important to discover the capabilities before applying the telemetry configuration (such as on-change subscription). Some of their content is part of the context for the streamed data.

This document covers only metadata about the collection context for the telemetry. The collected data is likely to be transformed into usable indicators for the network. The list of such transformation operations applied to the data is often called data lineage. Supplying the data lineage for the computed indicators is out of scope of this document.

This document enables retrieving the context in which a particular piece of data was collected, without having to access the platform that emitted the telemetry. This retrieval requires three elements: the time of data emission, the originating platform and the subscription through which the data arrived. The approach described in this document assumes that an underlying database records the history of both the data and the Data Manifest, enabling the data analyst to temporally match the data and the Data Manifest. Therefore, the approach presented here focuses on providing a way to match a platform and a subscription identifier to the collection context. This is consistent with most of the YANG modules for devices, which focus on describing the current state of the device, rather than the evolution of that state through time.

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.

This document reuses the term "Network Telemetry" as defined in [RFC9232].

Platform: equipment in the network able to produce telemetry data.

Data Manifest: The necessary data required to interpret telemetry information.

Platform Manifest: part of the Data Manifest that characterizes the platform producing the telemetry information.

Data Collection Manifest: part of the Data Manifest that characterizes how and when the telemetry information was metered.

Datapoint: an instance of data collected via Network Telemetry at a specific time.

Collector: software that receives the stream of telemetry.

3. Use Cases

3.1. Network Analytics

Streamed information from network platforms is used for network analytics, incident detection, and in the closed control loop for network automation. See [I-D.ietf-nmop-terminology] for definition of some of these terms. This streamed data can be stored in a time-series database or processed in a real-time streaming processor for further analysis.

As an example, a database could store a time series representing the evolution of a specific counter collected from a network platform. When analyzing the data, a network operator/data scientist must understand the context information for these data:

- * The counter definition, typically as defined in the YANG model.
- * The network platform vendor, model, and OS.
- * The collection parameters.

Characterizing the source used for producing the data (vendor, platform, and OS) is useful to complement the data. As an example, knowing the exact data source software specification might reveal a particularity in the observed data, explained by a specific bug, a specific bug fix, or simply a particular specific behavior. This is also necessary to ensure the reliability of the collected data. On top of that, for YANG-Push [RFC8641], it is crucial to know the set of YANG modules supported by the platform, along with their deviations. In some cases, there might even be some backwards incompatible changes in built-in modules (i.e., vendor proprietary modules) between one OS version to the next one. This information is captured by means of the Platform Manifest Section 5.

From a collection parameters point of view, the data scientists analyzing the collected data must know whether the counter was requested from the network platform as on-change or at specific cadence [RFC8641]. Indeed, an on-change collection explains why there is a single value as opposed to a time series. In case of

periodic collection, this exact cadence might not be observable in the time series. Indeed, this time series might report some values as 0 or might even omit some values. The reason for this behavior might be diverse: the network platform may be under stress, with a too small observation period, compared to the minimum-observed-period. Knowing the conditions under which a counter was collected and streamed (along with the platform details) helps drawing the informed conclusions. As an example, some platform might erroneously report a value of 0 for counters when the collection period is too short with respect to the capabilities of the platform. Without context, this value of 0 might lead to a wrong conclusion that the corresponding counter dropped to zero.

3.2. New Device Onboarding

When a new device is onboarded, operators must check that the new device streams data (e.g., with YANG-Push), that the Network Telemetry data is the right one, that the data is correctly collected at the data collection, and finally that the data can be analyzed (compared with other similar devices). For the last point, the Data Manifest, which must be linked to the data up to the collection and analytics system, contains the relevant information, notably in the case where the telemetry subscriptions are not configured by the telemetry collectors.

3.3. Data Mesh Principles in Networking

The concept behind the data mesh [DataMesh] are:

- * Domain Ownership: Architecturally and organizationally align business, technology, and analytical data, following the line of responsibility. The Data Mesh principles adopt the boundary of bounded context to individual data products where each domain is responsible for (and owns) its data and models.
- * Data as a Product: The "Domain" owners are responsible to provide the data in useful way (discoverable through a catalog, addressable with a permanent and unique address, understandable with well-defined semantics, trustworthy and truthful, self-describing for easy consumption, interoperable by supporting standards, secure, self-contained, etc.) and should treat consumers of that data as customers. It requires and relies on the "Domain Ownership" principle.
- * Self-serve Data Platform: This fosters the sharing of cross-domain data to create extra value.

- * **Federated Computational Governance:** Describes the operating model and approach to establishing global policies across a mesh of data products.

The most relevant concept for this document is the "Data as a Product" principle. The Data Manifest fulfills this principle as the two YANG data models, Platform Manifest and the Data Collection Manifest, along with the data, provide all the necessary information in a self-describing way for easy consumption.

4. The "ietf-yp-current-period" YANG module

Some platforms will adjust the collection period depending on their capabilities and current load. The YANG module in this section augments the "ietf-subscribed-notification" module to provide the "current-period" leaf. The value of this leaf indicates the current collection period which might be different from the configured collection period.

Figure 1 contains the YANG tree diagram [RFC8340] of the "ietf-yp-current-period" module.

module: ietf-yp-current-period

```
augment /sn:subscriptions/sn:subscription:
  +--ro current-period?   yp:centiseconds
```

Figure 1: YANG tree diagram for "ietf-yp-current-period" module

The code of the "ietf-yp-current" YANG module is given below.

```
<CODE BEGINS> file "ietf-yp-current-period@2025-02-21.yang"
module ietf-yp-current-period {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-yp-current-period";
  prefix yp-cp;

  import ietf-subscribed-notifications {
    prefix sn;
    reference
      "RFC 8639: A YANG Data Model for Subscriptions to
       Event Notifications";
  }
  import ietf-yang-push {
    prefix yp;
    // RFC Ed.: remove revision-date, needed here for datatracker
    // to properly validate the module, because the latest version
    // on the server is not the ratified one and validation fails.
  }
}
```

```
    revision-date 2019-09-09;
    reference
      "RFC 8641: Subscriptions to YANG Datastores.";
  }

organization
  "IETF OPSAWG (Operations and Management Area) Working Group";
contact
  "WG Web:    <https://datatracker.ietf.org/wg/opsawg/>
  WG List:    <mailto:opsawg@ietf.org>
  Author:     Benoit Claise    <mailto:benoit.claise@huawei.com>
  Author:     Jean Quilbeuf    <mailto:jean.quilbeuf@huawei.com>
  Author:     Diego R. Lopez    <diego.r.lopez@telefonica.com>
  Author:     Ignacio Dominguez <ignacio.dominguezmartinez@telefonica.com>
  Author:     Thomas Graf      <thomas.graf@swisscom.com>";
description
  "This module augments ietf-subscribed-notification and
  ietf-yang-push with the current-period leaf reporting the actual
  collection period, as opposed to the configured one.

  Copyright (c) 2025 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; see the
  RFC itself for full legal notices.  ";

revision 2025-02-21 {
  description
    "Initial revision";
  reference
    "RFC XXXX: A Data Manifest for Contextualized Telemetry Data";
}

augment "/sn:subscriptions/sn:subscription" {
  description
    "Adds current collection period";
  leaf current-period {
    when '../yp:periodic';
    type yp:centiseconds;
    config false;
    description
```



```
        "Period during two successive data collections, in the
        current state. Might differ from the configured period
        when the platform might increase the period
        automatically when it is overloaded.";
    }
}
}
<CODE ENDS>
```

5. Platform Manifest

5.1. Overview of the Model

Figure 2 contains the YANG tree diagram of the "ietf-platform-manifest module".

```

module: ietf-platform-manifest
+--ro platforms
  +--ro platform* [id]
    +--ro id string
    +--ro name? string
    +--ro vendor? string
    +--ro vendor-pen? uint32
    +--ro software-version? string
    +--ro software-flavor? string
    +--ro os-version? string
    +--ro os-type? string
    +--ro module-set* [name]
      +--ro name string
      +--ro module* [name]
        +--ro name yang:yang-identifier
        +--ro revision? revision-identifier
        +--ro namespace inet:uri
        +--ro location* inet:uri
        +--ro submodule* [name]
          +--ro name yang:yang-identifier
          +--ro revision? revision-identifier
          +--ro location* inet:uri
        +--ro feature* yang:yang-identifier
        +--ro deviation* -> ../../module/name
      +--ro import-only-module* [name revision]
        +--ro name yang:yang-identifier
        +--ro revision union
        +--ro namespace inet:uri
        +--ro location* inet:uri
        +--ro submodule* [name]
          +--ro name yang:yang-identifier
          +--ro revision? revision-identifier
          +--ro location* inet:uri
    +--ro schema* [name]
      +--ro name string
      +--ro module-set* -> ../../module-set/name
    +--ro datastore* [name]
      +--ro name ds:datastore-ref
      +--ro schema -> ../../schema/name

```

Figure 2: YANG tree diagram for ietf-platform-manifest module

The YANG module contains a list of Platform Manifests (in 'platforms/platform'), indexed by the identifier of the platform. That identifier should be defined by the network manager so that each platform emitting Network Telemetry has a unique identifier. There are several documents about managing the inventory of the network, e.g., [I-D.ietf-ivy-network-inventory-yang]. The platform identifier

should be the same as the identifier used in inventories or the 'node-id' in [RFC8345]. As an example, the identifier could be the 'sysName' from [RFC3418]. Selecting the correct identifier out of the existing works in IETF is out of scope for this document.

The scope of the "ietf-platform-manifest" module is the scope of the data collection, i.e., a given network, therefore it contains a collection of Platform Manifests, as opposed to the device scope, which would contain a single Platform Manifest. For a device-level implementation the same module is used, including a single entry in 'platforms/platform', corresponding to the local view of the network from the device point of view.

The Platform Manifest is characterized by a set of parameters ('name', 'software-version', 'software-flavor', 'os-version', and 'os-type') that are aligned with the YANG Catalog [I-D.clacla-netmod-model-catalog] so that the YANG Catalog could be used to retrieve the YANG modules a posteriori. The vendor of the platform can be identified via its name 'vendor' or its PEN number 'vendor-pen', as described in [RFC9371].

The Platform Manifest also includes the contents of the YANG Library [RFC8525]. That module set is particularly useful to retrieve the YANG modules associated to a subscription by analyzing the xpath filters or the subtree filters. Xpath filters are based on module names (see [RFC8639], description of leaf 'stream-xpath-filter', page 45). Subtree filters are based on namespaces.

5.2. "ietf-platform-manifest" YANG module

This section defines the "ietf-platform-manifest" YANG module.

```
<CODE BEGINS> file "ietf-platform-manifest@2025-02-21.yang"
module ietf-platform-manifest {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-platform-manifest";
  prefix p-mf;

  import ietf-yang-library {
    prefix yanglib;
    reference
      "RFC8525: YANG Library";
  }

  organization
    "IETF OPSAWG (Operations and Management Area) Working Group";
  contact
    "WG Web:    <https://datatracker.ietf.org/wg/opsawg/>
```



```
}
leaf vendor {
  type string {
    length "1..1023";
  }
  description
    "Organization that implements that platform.";
}
leaf vendor-pen {
  type uint32;
  description
    "Vendor's registered Private Enterprise Number";
  reference
    "RFC9371: Registration Procedures for Private Enterprise
    Numbers (PENs)";
}
leaf software-version {
  type string {
    length "1..1023";
  }
  description
    "Name of the version of software. With respect to most
    network device appliances, this will be the operating system
    version. But for other YANG module implementation, this
    would be a version of appliance software. Ultimately, this
    should correspond to a version string that will be
    recognizable by the consumers of the platform.";
}
leaf software-flavor {
  type string {
    length "1..1023";
  }
  description
    "A variation of a specific version where YANG model support
    may be different. Depending on the vendor, this could be a
    license, additional software component, or a feature set.";
}
leaf os-version {
  type string {
    length "1..1023";
  }
  description
    "Version of the operating system using this module. This is
    primarily useful if the software implementing the module is
    an application that requires a specific operating system
    version.";
}
leaf os-type {
```

```
    type string {
      length "1..1023";
    }
    description
      "Type of the operating system using this module. This is
      primarily useful if the software implementing the module is
      an application that requires a specific operating system
      type.";
  }
}

container platforms {
  config false;
  description
    "Top container including all platforms in scope. If this model
    is hosted on a single device, it should contain a single entry
    in the list. At the network level, it should contain an entry
    for every monitored platform.";
  list platform {
    key "id";
    description
      "Contains information about the platform that allows
      identifying and understanding the individual data collection
      information.";
    leaf id {
      type string {
        length "1..1023";
      }
      description
        "Identifies a given platform on the network, for instance
        the 'sysName' of the platform. The 'id' has to be unique
        within the network scope at every point in time. The same
        id can point to different platform if they are not
        simultaneously part of the network, e.g., when a device
        associated to a particular id is replaced.";
    }
    uses platform-details;
    uses yanglib:yang-library-parameters;
  }
}
}
<CODE ENDS>
```

6. Data Collection Manifest

This section is non-normative.

6.1. Overview of the Model

Figure 3 contains the YANG tree diagram [RFC8340] of the "example-collection-manifest" module. The module relies upon the YANG Schema mount [RFC8528] to reuse existing YANG modules describing the current data collection status. This module is an example, i.e. non-normative, as YANG Schema mount does not support design-time schema mount. Appendix C explains how the YANG tree is obtained.

```

module: example-collection-manifest
  +--ro data-collections
    +--mp data-collection* [platform-id]
      +--ro platform-id      -> /p-mf:platforms/p-mf:platform/p-mf:id
      +--ro streams/
        +--ro stream* [name]
          +--ro name          string
          +--ro description?  string
      +--ro filters/
        +--ro stream-filter* [name]
          +--ro name          string
          +--ro (filter-spec)?
            +--:(stream-subtree-filter)
            +--:(stream-xpath-filter)
              +--ro stream-xpath-filter?  yang:xpath1.0
              {xpath}?
        +--ro selection-filter* [filter-id]
          +--ro filter-id      string
          +--ro (filter-spec)?
            +--:(datastore-subtree-filter)
            +--:(datastore-xpath-filter)
              +--ro datastore-xpath-filter?  yang:xpath1.0
              {sn:xpath}?
      +--ro subscriptions/
        +--ro subscription* [id]
          +--ro id              subscription-id
          +--ro (target)
            +--:(stream)
              +--ro (stream-filter)?
                +--:(by-reference)
                  +--ro stream-filter-name
                  stream-filter-ref
                +--:(within-subscription)
                  +--ro (filter-spec)?
                    +--:(stream-subtree-filter)
                    +--:(stream-xpath-filter)
                      +--ro stream-xpath-filter?
                      yang:xpath1.0 {xpath}?
                  +--ro stream
                  stream-ref

```

```

|   +--:(datastore)
|   |   +--ro datastore          identityref
|   |   +--ro (selection-filter)?
|   |   |   +--:(by-reference)
|   |   |   |   +--ro selection-filter-ref
|   |   |   |   |   selection-filter-ref
|   |   +--:(within-subscription)
|   |   |   +--ro (filter-spec)?
|   |   |   |   +--:(datastore-subtree-filter)
|   |   |   |   +--:(datastore-xpath-filter)
|   |   |   |   |   +--ro datastore-xpath-filter?
|   |   |   |   |   |   yang:xpath1.0 {sn:xpath}?
+--ro stop-time?          yang:date-and-time
+--ro encoding?           encoding
+--ro receivers
|   +--ro receiver* [name]
|   |   +--ro name              string
|   |   +--ro sent-event-records?
|   |   |   yang:zero-based-counter64
|   |   +--ro excluded-event-records?
|   |   |   yang:zero-based-counter64
|   |   +--ro state              enumeration
+--ro (update-trigger)?
|   +--:(periodic)
|   |   +--ro periodic!
|   |   |   +--ro period          centiseconds
|   |   |   +--ro anchor-time?    yang:date-and-time
|   +--:(on-change) {on-change}?
|   |   +--ro on-change!
|   |   |   +--ro dampening-period? centiseconds
|   |   |   +--ro sync-on-start?   boolean
|   |   |   +--ro excluded-change* change-type
+--ro current-period?      yp:centiseconds

```

Figure 3: YANG tree diagram for example-collection-manifest module

The 'data-collections' container contains the information related to each YANG-Push subscription. As for the Platform Manifest, these subscriptions are indexed by the 'platform-id', so that all subscriptions in the network can be represented at the network level without any conflict.

As most of the information related to YANG-push subscription [RFC8639] and [RFC8641] is stored in the "ietf-yang-push" module, these modules are mounted. These modules have a part common to all subscriptions of the platform, stored in the 'streams' and 'filters' containers. The information about subscriptions themselves are stored in the 'subscriptions/subscription' list, indexed by a subscription identifier.

In the subscription object, the 'current-period' indicates the period currently used between two updates. That leaf can only be present when the subscription is periodic. The current period might differ from the requested period if the platform implements a mechanism to increase the collection period when it is overloaded. Having the current period information is crucial to understand if telemetry is missing because of a bug or a packet loss or simply because it was dynamically adjusted by the platform.

The 'current-period' data node is added by the module 'ietf-data-collection-manifest-statistics' presented in Section 4. This module augments the subscription list from the module 'ietf-subscribed-notifications'. It is mounted as well via the YANG Schema Mount mechanism. The module for the Data Collection Manifest is presented in Section 6.2.

6.2. The "example-collection-manifest" YANG module

This section includes the code of the "example-collection-manifest" YANG module. Additionally, it defines the extension data file for YANG schema mount. The Data Collection Manifest should conform to the model obtained by combining these two specifications.

```
module example-collection-manifest {
  yang-version 1.1;
  namespace "http://example.com/example-data-collection-manifest";
  prefix ex-d-mf;

  import ietf-platform-manifest {
    prefix p-mf;
    reference
      "RFC XXXX: A Data Manifest for Contextualized Telemetry
      Data";
  }
  import ietf-yang-schema-mount {
    prefix yangmnt;
    reference
      "RFC8528: YANG Schema Mount";
  }
}
```

organization

"IETF OPSAWG (Operations and Management Area) Working Group";

contact

"WG Web: <<https://datatracker.ietf.org/wg/opsawg/>>

WG List: <<mailto:opsawg@ietf.org>>

Author: Benoit Claise <<mailto:benoit.claise@huawei.com>>

Author: Jean Quilbeuf <<mailto:jean.quilbeuf@huawei.com>>

Author: Diego R. Lopez <diego.r.lopez@telefonica.com>

Author: Ignacio Dominguez
<ignacio.dominguezmartinez@telefonica.com>

Author: Thomas Graf <thomas.graf@swisscom.com>;

description

"This module describes the context of data collection from a given network element. The contents of this model must be streamed along with the data streamed from the network element so that the context of the data collection can be retrieved later.

This module must be completed with
ietf-platform-manifest
to capture the whole context of a data collection session.

Copyright (c) 2025 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; see the RFC itself for full legal notices. ";

revision 2025-02-21 {

description

"Initial revision";

reference

"RFC XXXX: A Data Manifest for Contextualized Telemetry Data";

}

container data-collections {

config false;

description

"Contains the configuration and statistics for the collected data, per node in the network.";

list data-collection {

key "platform-id";

```

description
  "Defines the information for each collected object.";
leaf platform-id {
  type leafref {
    path "/p-mf:platforms/p-mf:platform/p-mf:id";
  }
  description
    "Identifier of the platform collecting the data. This
    identifier is the same as the one in the platform
    manifest.";
}
yangmnt:mount-point "yang-push-collection" {
  description
    "This mount point must mount the following modules and
    their dependencies:
    * ietf-subscribed-notifications
    * ietf-yang-push
    * ietf-yp-current-period.
    This mount point must not mount any other modules.";
  reference
    "RFC8639: Subscription to YANG Notifications
    RFC8641: Subscription to YANG Notifications for datastore
    updates";
}
}
}
}

```

7. Data Manifest and the Collected Data

This section focuses on associating the collected data to the Data Manifest. As this document specifically focuses on giving context on data collected via Network Telemetry, it is assumed that a Network Telemetry system is available. Another premise of this document is the storage of the collected data into a database for later exploitation. This document assumes that such a database exists and can be used for storing the Data Manifest.

7.1. Collecting the Data Manifest

The Data Manifest **MUST** be streamed and stored along with the collected data. In case the collected data are moved to a different place (typically a database), the companion Data Manifest **MUST** follow the collected data. Storing the collected data without the companion Data Manifest might prevent the correct interpretation of the collected data. The Data Manifest **MUST** be updated when the Data Manifest information changes, for example, when a router is upgraded, when a new Network Telemetry subscription is configured, or when the

Network Telemetry subscription parameters change. The Data Manifest can itself be considered as a time series, and stored in a similar fashion to the collected data.

This document recommends reusing existing Network Telemetry systems (in-band approach) to lower the efforts for implementing this approach. To enable a platform supporting Network Telemetry to also support the Data Manifest, it is sufficient that this platform supports the models from Sections 5 and 6. The collection of the Data Manifest MUST be explicitly configured by the collector by requesting the relevant subscriptions. These subscriptions MUST include the Platform Manifest and the Data Collection Manifest, possibly limited to the subscriptions for which the context needs to be retrieved a posteriori. Appendix B shows how the in-band approach would work while storing to a time series database.

Each type of manifest has its own rough frequency update, i.e., at reboot for the Platform Manifest and when subscriptions are modified for the Data Collection Manifest. The Data Manifest SHOULD be streamed with the YANG-Push on-change feature [RFC8641] (also called event-driven telemetry) whenever possible.

A Platform Manifest is likely to remain the same until the platform is updated. Thus, the Platform Manifest only needs to be collected once per streaming session and updated after a platform reboot. The "subscription-terminated" (Section 2.7.3 of [RFC8639]) will indicate to the collector that the platform rebooted. The collector MUST then collect the potential update of the Platform Manifest on re-establishment of the subscription. Using the on-change feature enables to capture dynamic changes to the Platform Manifest as well, if any.

Regarding the Data Manifest, the elements common to all subscriptions, such as the stream definitions and the common filters might be updated less frequently than the subscriptions. Relying on YANG-Push on-change feature enables keeping an up-to-date version of the Data Collection Manifest.

The underlying time series database should accommodate the various rates at which different parts of the Data Manifest are updated. In particular, storing the Platform Manifest should be optimized to avoid duplicating repeated content and only storing a new version when there is a change in the manifest.

7.2. Mapping Collected Data to the Data Manifest

As explained in the introduction, three elements are necessary to identify the Data Manifest associated to a datapoint:

- * the time at which the data was sent from the device,
- * the originating platform sending the data, and
- * the identifier of the subscription that produced the data.

These elements can be either known to the collector, if it is the one configuring the collection, or retrieved via dedicated headers as proposed, e.g., in [I-D.netana-netconf-notif-envelope]. To enable a posteriori retrieval of the Data Manifest associated to a datapoint, the collector MUST keep the subscription identifier and platform identifier in the metadata of the collected values.

With these three elements, to retrieve the Data Manifest from a datapoint, the following happens:

- * The subscription identifier, platform identifier and timestamp of the data are retrieved from the datapoint metadata
- * The Platform Manifest for that datapoint is obtained by looking up the latest version before the timestamp matching the platform identifier.
- * The Data Collection Manifest for that datapoint is obtained by looking up the latest version before the timestamp matching the platform identifier and the subscription identifier.

The reliability of the collection of the Data Manifest is the same as the reliability of the data collection itself, since the Data Manifest is like any other data.

8. Operational Considerations

The Data Manifest is partially standardized in this document since only the Platform Manifest has a normative YANG module. Section 6.1 explains why the Data Collection Manifest cannot be normative at the current time. In practice, the Data Collection Manifest might be difficult to implement due to the dependency on YANG Schema Mount and the lack of normative YANG module. In that case, the solution proposed in the next paragraph covers the creation of the Data Collection Manifest.

It is expected that the Data Manifest is streamed directly from the network equipment, along with YANG-Push [RFC8641] data. However, if the equipment Network Telemetry does not yet support the YANG modules from the Data Manifest specified in this document, the telemetry collector could populate the Data Manifest from available information collected from the platform. This latter option requires efforts on

the Network Telemetry data collection side, as the information gathered in the Data Manifest proposed in this document could be scattered among various standard and vendor-specific YANG modules [RFC8199], that depend on the platform.

The current approach relies on the hostname for identifying platforms, which identifies a role rather than a particular piece of hardware. As a consequence, there is no simple solution for obtaining the Data Manifest of a given piece of hardware reused for different roles in the same network. In general, the problem of having consistent metadata among various protocols is still an open problem as detailed in Section 4.10 of [I-D.boucadair-nmop-rfc3535-20years-later].

9. Example

Figure 4 shows an example of both a Platform Manifest and corresponding Data Collection Manifests. The list of YANG modules in the 'yang-library' container is kept empty for brevity.

```
{
  "ietf-platform-manifest:platforms": {
    "platform": [
      {
        "id": "PE1",
        "name": "PE1",
        "vendor": "ACME",
        "vendor-pen": 32473,
        "software-version": "3.14",
        "os-version": "2.79",
        "os-type": "ACME OS"
      }
    ]
  },
  "example-collection-manifest:data-collections": {
    "data-collection": [
      {
        "platform-id": "PE1",
        "ietf-subscribed-notifications:subscriptions": {
          "subscription": [
            {
              "id": 4242,
              "ietf-yang-push:datastore":
                "ietf-datastores:operational",
              "ietf-yang-push:datastore-xpath-filter":
                "/ietf-interfaces:interfaces/interface/enabled",
              "ietf-yang-push:on-change": {}
            }
          ]
        }
      }
    ]
  }
}
```

```

    "receivers": {
      "receiver": [
        {
          "name": "yp-collector",
          "state": "active"
        }
      ]
    },
    {
      "id": 4243,
      "ietf-yang-push:datastore":
        "ietf-datastores:operational",
      "ietf-yang-push:datastore-xpath-filter":
"/ietf-interfaces:interfaces/interface/statistics/in-octets",
      "ietf-yang-push:periodic": {
        "period": 10000
      },
      "ietf-yp-current-period:current-period": 20000,
      "receivers": {
        "receiver": [
          {
            "name": "yp-collector",
            "state": "active"
          }
        ]
      }
    }
  ],
}

```

Figure 4: Example of Data Manifest

Figure 4 contains the Data Collection Manifest for two XPath subscriptions. With the Data Collection Manifest for the first one, with subscription identifier 4242, the exact semantics of the collected path, here the administrative status of the network interfaces, can be obtained by looking up the module in the yang-library of the corresponding Platform Manifest, to obtain the exact revision of ietf-interfaces used at collection time. Also, the "on-change" container indicates that data will be sent only if there is a change, thus not receiving data indicates that the administrative status of the interface did not change.

The other example of Data Collection Manifest, with subscription identifier 4243, shows how a periodic subscription is reported. In that example, the 'current-period' indicates that the requested period of 10s (1000 centiseconds) could not be attained and is now of 20s, for instance because the device is overloaded.

Appendix D gives the command line for validating this example using [yanglint].

10. Security Considerations

This section is modeled after the template described in Section 3.7 of [I-D.ietf-netmod-rfc8407bis].

The "ietf-platform-manifest" module defines a data model that is designed to be accessed via YANG-based management protocols, such as NETCONF [RFC6241] and RESTCONF[RFC8040]. These protocols have to use a secure transport layer (e.g., SSH [RFC6242], TLS [RFC8446] and QUIC [RFC9000]) and have to use mutual authentication.

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.

Some of the readable data nodes in this YANG module may be considered sensitive or vulnerable in some network environments. It is thus important to control read access (e.g., via get, get-config, or notification) to these data nodes. Specifically, the following subtrees and data nodes have particular sensitivities/vulnerabilities:

- * platforms/platform contains details about the platform that an attacker could use to find the known vulnerabilities of the platform.

The "ietf-yp-current-period" module defines a data model that is designed to be accessed via YANG-based management protocols, such as NETCONF [RFC6241] and RESTCONF[RFC8040]. These protocols have to use a secure transport layer (e.g., SSH [RFC6242], TLS [RFC8446] and QUIC [RFC9000]) and have to use mutual authentication.

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.

Some of the readable data nodes in this YANG module may be considered sensitive or vulnerable in some network environments. It is thus important to control read access (e.g., via get, get-config, or notification) to these data nodes. Specifically, the following subtrees and data nodes have particular sensitivities/vulnerabilities:

There are no particularly sensitive readable data nodes.

As the present approach reuses an existing telemetry system, the security considerations lie with the new content divulged in the new manifests. Appropriate access control filters must be associated to the corresponding leafs and containers, as well as the databases storing them.

The integrity and provenance of the data of the collection manifest can be ensured by a signing mechanism such as [I-D.lopez-opsawg-yang-provenance].

11. IANA Considerations

RFC Ed.: replace XXXX with actual RFC number and remove this note.

IANA is requested to register the following URIs in the "ns" subregistry within the "IETF XML Registry" [RFC3688]:

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

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

IANA is requested to register the following YANG modules in the "YANG Module Names" subregistry [RFC6020] within the "YANG Parameters" registry.

Name: ietf-platform-manifest
Maintained by IANA? N
Namespace: urn:ietf:params:xml:ns:yang:ietf-platform-manifest
Prefix: p-mf
Reference: RFC XXXX

Name: ietf-yp-current-period
Maintained by IANA? N
Namespace: urn:ietf:params:xml:ns:yang:ietf-yp-current-period
Prefix: yp-cp
Reference: RFC XXXX

12. Contributors

13. Open Issues

This section is to be removed before publishing as an RFC.

- * Do we want to handle the absence of values, i.e. add information about missed collection or errors in the collection context ? It could also explain why some values are missing. On the other hand, this might also be out scope. CLOSED: the goal of the manifest is to be able to detect miscollection a posteriori. Assurance of the metric collection is out of scope and could be done via an external mechanism such as SAIN.
- * Henk: how does this interact with SBOM effort? CLOSED: SBOM is another kind of manifest, we are focusing here on data collection.
- * What is the link with the RFC8345 NodeId and IVY? CLOSED: added text.
- * Handling of deletion in [I-D.kll-yang-label-tsdb]. CLOSED: out of scope

14. 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>>.
- [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>>.
- [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>>.
- [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>>.
- [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>>.
- [RFC8525] Bierman, A., Bjorklund, M., Schoenwaelder, J., Watsen, K., and R. Wilton, "YANG Library", RFC 8525, DOI 10.17487/RFC8525, March 2019, <<https://www.rfc-editor.org/info/rfc8525>>.
- [RFC8528] Bjorklund, M. and L. Lhotka, "YANG Schema Mount", RFC 8528, DOI 10.17487/RFC8528, March 2019, <<https://www.rfc-editor.org/info/rfc8528>>.
- [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>>.

15. Informative References

- [DataMesh] "Datamesh Architecture", <<https://www.datamesh-architecture.com/>>.
- [I-D.boucadair-nmop-rfc3535-20years-later] Boucadair, M., Contreras, L. M., de Dios, O. G., Graf, T., Rahman, R., and L. Tailhardat, "RFC 3535, 20 Years Later: An Update of Operators Requirements on Network Management

Protocols and Modelling", Work in Progress, Internet-Draft, draft-boucadair-nmop-rfc3535-20years-later-08, 12 May 2025, <<https://datatracker.ietf.org/doc/html/draft-boucadair-nmop-rfc3535-20years-later-08>>.

[I-D.clacla-netmod-model-catalog]

Clarke, J. and B. Claise, "YANG module for yangcatalog.org", Work in Progress, Internet-Draft, draft-clacla-netmod-model-catalog-03, 3 April 2018, <<https://datatracker.ietf.org/doc/html/draft-clacla-netmod-model-catalog-03>>.

[I-D.claise-netconf-metadata-for-collection]

Claise, B., Nayyar, M., and A. R. Sesani, "Per-Node Capabilities for Optimum Operational Data Collection", Work in Progress, Internet-Draft, draft-claise-netconf-metadata-for-collection-03, 25 January 2022, <<https://datatracker.ietf.org/doc/html/draft-claise-netconf-metadata-for-collection-03>>.

[I-D.ietf-ivy-network-inventory-yang]

Yu, C., Belotti, S., Bouquier, J., Peruzzini, F., and P. Bedard, "A Base YANG Data Model for Network Inventory", Work in Progress, Internet-Draft, draft-ietf-ivy-network-inventory-yang-11, 14 October 2025, <<https://datatracker.ietf.org/doc/html/draft-ietf-ivy-network-inventory-yang-11>>.

[I-D.ietf-netmod-rfc8407bis]

Bierman, A., Boucadair, M., and Q. Wu, "Guidelines for Authors and Reviewers of Documents Containing YANG Data Models", Work in Progress, Internet-Draft, draft-ietf-netmod-rfc8407bis-28, 5 June 2025, <<https://datatracker.ietf.org/doc/html/draft-ietf-netmod-rfc8407bis-28>>.

[I-D.ietf-nmop-terminology]

Davis, N., Farrel, A., Graf, T., Wu, Q., and C. Yu, "Some Key Terms for Network Fault and Problem Management", Work in Progress, Internet-Draft, draft-ietf-nmop-terminology-23, 18 August 2025, <<https://datatracker.ietf.org/doc/html/draft-ietf-nmop-terminology-23>>.

[I-D.kll-yang-label-tsdb]

Larsson, K., "Mapping YANG Data to Label-Set Time Series", Work in Progress, Internet-Draft, draft-kll-yang-label-tsdb-00, 18 October 2023, <<https://datatracker.ietf.org/doc/html/draft-kll-yang-label-tsdb-00>>.

[I-D.lopez-opsawg-yang-provenance]

Lopez, D., Pastor, A., Feng, A. H., Pirez, A. M., Birkholz, H., and S. Garcia, "Applying COSE Signatures for YANG Data Provenance", Work in Progress, Internet-Draft, draft-lopez-opsawg-yang-provenance-07, 23 April 2025, <<https://datatracker.ietf.org/doc/html/draft-lopez-opsawg-yang-provenance-07>>.

[I-D.netana-netconf-notif-envelope]

Feng, A. H., Francois, P., Graf, T., and B. Claise, "Extensible YANG Model for YANG-Push Notifications", Work in Progress, Internet-Draft, draft-netana-netconf-notif-envelope-02, 28 January 2025, <<https://datatracker.ietf.org/doc/html/draft-netana-netconf-notif-envelope-02>>.

[RFC3418] Presuhn, R., Ed., "Management Information Base (MIB) for the Simple Network Management Protocol (SNMP)", STD 62, RFC 3418, DOI 10.17487/RFC3418, December 2002, <<https://www.rfc-editor.org/info/rfc3418>>.

[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>>.

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

[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>>.

[RFC8343] Bjorklund, M., "A YANG Data Model for Interface Management", RFC 8343, DOI 10.17487/RFC8343, March 2018, <<https://www.rfc-editor.org/info/rfc8343>>.

- [RFC8345] Clemm, A., Medved, J., Varga, R., Bahadur, N., Ananthakrishnan, H., and X. Liu, "A YANG Data Model for Network Topologies", RFC 8345, DOI 10.17487/RFC8345, March 2018, <<https://www.rfc-editor.org/info/rfc8345>>.
- [RFC9000] Iyengar, J., Ed. and M. Thomson, Ed., "QUIC: A UDP-Based Multiplexed and Secure Transport", RFC 9000, DOI 10.17487/RFC9000, May 2021, <<https://www.rfc-editor.org/info/rfc9000>>.
- [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>>.
- [RFC9232] Song, H., Qin, F., Martinez-Julia, P., Ciavaglia, L., and A. Wang, "Network Telemetry Framework", RFC 9232, DOI 10.17487/RFC9232, May 2022, <<https://www.rfc-editor.org/info/rfc9232>>.
- [RFC9371] Baber, A. and P. Hoffman, "Registration Procedures for Private Enterprise Numbers (PENs)", RFC 9371, DOI 10.17487/RFC9371, March 2023, <<https://www.rfc-editor.org/info/rfc9371>>.
- [yanglint] "Yanglint", <<https://github.com/CESNET/libyang>>.

Appendix A. Changes between revisions

This section is to be removed before publishing as an RFC.

v09 -> v10

- * Address Mahesh comments

v08 -> v09

- * Removed unused ref, move 9232 to informational
- * Answer Joe's comments in Operational Consideration (which is re-renamed by the way)

v07 -> v08

- * Address comments from Thomas

v06 -> v07

- * Operational +(and management) considerations (draft-opsarea-rfc5706bis)

- * Make current period config false

- * Explicit that data collection is non-normative

- * Adjust security section to RFC8407bis new template

- * Other comments from Med

v05 -> v06

- * Example can be validated using yanglint

- * Applied details comments from Joe and Med

- * Making the "current-period" update more generic and mentioning it in the introduction

- * Section 7 (previously 5) reworked to clarify how data manifest is collected and retrieved from a datapoint

- * Remove use of YANG schema mount for the platform manifest and change data collection manifest to example

v04 -> v05

- * Remove references to full-include draft, use schema mount.

- * Explain link with schema node id

v03 -> v04

- * State that data lineage is out of scope

- * Replace copy-pasted version of the modules with schema mount version, use full-embed for the "real" one

- * Schema mount version is the fallback plan if full:embed is not there fast enough.

- * Update examples accordingly

v02 -> v03

- * Explicit that modules are network (Controller) level

- * InfluxDB example changed to TSDB example aligned with [I-D.kll-yang-label-tsdb]
 - * Minor edits i.e. network element -> platform , object -> data node
- v01 -> v02
- * Updated example with latest version of the model.
- v00 (WG adoption) - v01
- * Solve integrity issue by delegating to [I-D.lopez-opsawg-yang-provenance].
- v05 -> v06
- * Remove YANG packages
 - * Switch YANG models from device view to network view
 - * Add PEN number to identify vendors
 - * Intro rewritten with uses cases
 - * Added an "Operational Considerations" section
 - * Switch from MDT to YANG-push
- v04 -> v05
- * First version of example scenario
 - * Updated affiliation
 - * Updated YANG module names to ietf-platform-manifest and ietf-data-collection-manifest
 - * Unify used terms as defined in the terminology section
 - * Replaced 'device' with 'platform'
 - * Split Section 5 into two sections for better readability
- v03 -> v04
- * Fix xym error
 - * Moved terminology after introduction

- * Clarified the role of the module

v02 -> v03

- * Add when clause in YANG model
- * Fix validation errors on YANG modules
- * Augment YANG library to handle semantic versioning

v01 -> v02

- * Alignment with YANGCatalog YANG module: name, vendor
- * Clarify the use of YANG instance file
- * Editorial improvements

v00 -> v01

- * Adding more into data platform: yang packages, whole yanglib module to specify datastores
- * Setting the right type for periods: int64 -> uint64
- * Specify the origin datastore for mdt subscription
- * Set both models to config false
- * Applying text comments from Mohamed Boucadair
- * Adding an example of data-manifest file
- * Adding rationale for reusing telemetry system for collection of the manifests
- * Export manifest with on change telemetry as opposed to YANG instance file

v00

- * Initial version

Appendix B. An Example of Use Based on MDT

In this example, the goal is to collect the administrative status and number of received bytes for the interfaces of a fictional ACME device, and store the result in a time-series database. The metrics are collected using YANG-Push, which is configured by specifying their XPaths and when they should be collected (periodically or on-change). More precisely, the Xpaths to collect are "ietf-interfaces:interfaces/interface/enabled" on every change and "ietf-interfaces:interfaces/interface/statistics/in-octets" every 100 milliseconds. The paths here are referring to the YANG module from [RFC8343]. The configuration of YANG push is out of scope for this document. Since they don't have the same trigger, each of the path must be collected in its own subscription. Figure 5 presents an example for such a collection.

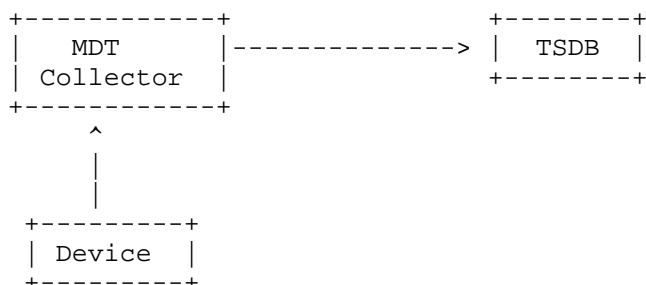


Figure 5: Example of Collection From a Device to a TSDB

In the scenario depicted in Figure 5, the collector receives YANG-push data from the device and stores it into a time series database (TSDB). This section first presents a version without Data Manifest and then how to enrich it with the Data Manifest.

Examples rely on the notation from [I-D.kll-yang-label-tsdb] to represent how the data is stored in the TSDB. Without the Data Manifest, the result of the collection would be stored as showed in Figure 6. The "host" label indicates the devices from which the data is collected and the YANG keys are included as well. Here the interface "eth0" is enabled and received 1234 octets. In that case, the value is stored, without any way to know how the value was obtained.

```
* Metric: interfaces_interface_enabled
* Value: True
* Labels:
  - host: "PE1"
  - interfaces_interface_name: "eth0"
--
* Metric: interfaces_interface_statistics_in_octets
* Value: 1234
* Labels:
  - host: "PE1"
  - interfaces_interface_name: "eth0"
```

Figure 6: Storing Datapoints without Data Manifest

An option for keeping the Data Manifest with the data is to store it directly into the TSDB. In that case, the collector can subscribe to the data exported by the module presented in this document and store it as other metrics. For the Platform Manifest, assuming the platform identifier is "PE1", the collector subscribes to the path "ietf-platform-manifest:platforms/platform[id=PE1]". For the Data Collection Manifests, the collector subscribes to the path "ietf-data-collection-manifest:data-collections/data-collection[platform-id="PE1"]/yang-push-collection/subscriptions/subscription[id=X]" where X is the subscription identifier of existing subscriptions. With the approach from [I-D.kll-yang-label-tsdb], the corresponding subtrees would be split into a set of datapoints, one per leaf. Figure 7 shows two examples of storing leaves in a TSDB. The first leaf is the vendor PEN number, which is part of the Platform Manifest. The second leaf is the Xpath filter used for subscription to the interface status.

```
* Metric: platforms_platform_vendor_pen
* Value: 32473
* Labels:
  - host: "PE1"
  - platforms_platform_id: "PE1"
--
* Metric: data_collections_data_collection_yang_push_collection_
      subscriptions_subscription_datastore_xpath_filter
* Value: "ietf-interfaces:interfaces/interface/enabled"
* Labels:
  - host: "PE1"
  - data_collections_data_collection_platform_id: "PE1"
  - data_collections_data_collection_yang_push_collection_
      subscriptions_subscription_id: 4242
```

Figure 7: Example of storing Platform and Data Collection Manifest: Vendor PEN and Xpath filter.

In the labels, the "host" might be different from the "platforms_platform_id" in case the collector is the one assembling it, i.e. for devices that do not support the Data Manifest. In that case, the value of this label could be the hostname of the collector. The host value does not matter for retrieving the Data Manifest as the platform identifier is the meaningful field.

In this example, retrieving the Platform Manifest associated to a collected datapoint is done by looking for datapoints that have the label "platforms_platform_id" equal to the value of the host for that collected datapoint. In order to link a datapoint with the corresponding Data Collection Manifest, an additional label for the subscription identifier is required. For instance, the same datapoints as in Figure 6 could be stored as in Figure 8.

```
* Metric: interfaces_interface_enabled
* Value: True
* Labels:
  - host: "PE1"
  - interfaces_interface_name: "eth0"
  - data_collections_data_collection_yang_push_subscriptions_
    subscription_id: 4242
--
* Metric: interfaces_interface_statistics_in_octets
* Value: 1234
* Labels:
  - host: "PE1"
  - interfaces_interface_name: "eth0"
  - data_collections_data_collection_yang_push_subscriptions_
    subscription_id: 4243
```

Figure 8: Storing datapoints with information to retrieve the Data Manifest

From the "interfaces_interface_enabled" datapoint, one can retrieve the corresponding Data Collection Manifest by looking for datapoints that have the label data_collections_data_collection_yang_push_collection_subscriptions_subscription_id equal to 4242.

Various optimizations could be done, such as relying on on-change subscription to modify only the leaves that changed. In that way, the amount of data needed for updating and storing the Data Manifest in the TSDB would be limited.

Appendix C. Generating YANG Tree Diagrams

This section provides the files needed to generate the YANG tree diagram [RFC8340] from Figure 3. The diagram was obtained using yanglint [yanglint] version 2.1.80, using the YANG Schema Mount [RFC8528]. It was manually edited to remove parts irrelevant to this document such as data nodes from imported modules, notifications and RPCs.

In order to get a tree diagram involving YANG Schema Mount with yanglint, two data files are required, in addition to the YANG module, its dependencies and the YANG modules to be mounted. The first required file the extension data, containing the YANG library to use at the mount point, this file is provided below as "data-collection-extension-data.xml". The second required file is the YANG library to use at the top-level context, this file is provided below as "data-collection-toplevel-yanglib.xml". The following command was used to obtain the YANG Tree diagram (before manual edition).

```
yanglint -f tree \  
-x data-collection-extension-data.xml \  
-Y data-collection-toplevel-yanglib.xml \  
example-collection-manifest@2025-02-21.yang
```

```
<CODE BEGINS> file "data-collection-extension-data.xml"  
<yang-library xmlns="urn:ietf:params:xml:ns:yang:ietf-yang-library"  
  xmlns:ds="urn:ietf:params:xml:ns:yang:ietf-datastores">  
  <module-set>  
    <name>mountee-set</name>  
    <module>  
      <name>ietf-subscribed-notifications</name>  
      <revision>2019-09-09</revision>  
      <namespace>  
        urn:ietf:params:xml:ns:yang:ietf-subscribed-notifications  
      </namespace>  
      <feature>xpath</feature>  
    </module>  
    <module>  
      <name>ietf-yang-push</name>  
      <revision>2019-09-09</revision>  
      <namespace>  
        urn:ietf:params:xml:ns:yang:ietf-yang-push  
      </namespace>  
      <feature>on-change</feature>  
    </module>  
    <module>  
      <name>ietf-yp-current-period</name>  
      <revision>2025-02-21</revision>
```

```
<namespace>
  urn:ietf:params:xml:ns:yang:ietf-yp-current-period
</namespace>
</module>
<module>
  <name>ietf-datastores</name>
  <revision>2018-02-14</revision>
  <namespace>
    urn:ietf:params:xml:ns:yang:ietf-datastores
  </namespace>
</module>
<module>
  <name>ietf-yang-library</name>
  <revision>2019-01-04</revision>
  <namespace>
    urn:ietf:params:xml:ns:yang:ietf-yang-library
  </namespace>
</module>
<import-only-module>
  <name>ietf-inet-types</name>
  <revision>2013-07-15</revision>
  <namespace>
    urn:ietf:params:xml:ns:yang:ietf-inet-types
  </namespace>
</import-only-module>
<import-only-module>
  <name>ietf-interfaces</name>
  <revision>2018-02-20</revision>
  <namespace>
    urn:ietf:params:xml:ns:yang:ietf-interfaces
  </namespace>
</import-only-module>
<import-only-module>
  <name>ietf-ip</name>
  <revision>2018-02-22</revision>
  <namespace>
    urn:ietf:params:xml:ns:yang:ietf-ip
  </namespace>
</import-only-module>
<import-only-module>
  <name>ietf-netconf-acm</name>
  <revision>2018-02-14</revision>
  <namespace>
    urn:ietf:params:xml:ns:yang:ietf-netconf-acm
  </namespace>
</import-only-module>
<import-only-module>
  <name>ietf-network-instance</name>
```

```
<revision>2019-01-21</revision>
<namespace>
  urn:ietf:params:xml:ns:yang:ietf-network-instance
</namespace>
</import-only-module>
<import-only-module>
  <name>ietf-restconf</name>
  <revision>2017-01-26</revision>
  <namespace>
    urn:ietf:params:xml:ns:yang:ietf-restconf
  </namespace>
</import-only-module>
<import-only-module>
  <name>ietf-yang-patch</name>
  <revision>2017-02-22</revision>
  <namespace>
    urn:ietf:params:xml:ns:yang:ietf-yang-patch
  </namespace>
</import-only-module>
<import-only-module>
  <name>ietf-yang-types</name>
  <revision>2023-01-23</revision>
  <namespace>
    urn:ietf:params:xml:ns:yang:ietf-yang-types
  </namespace>
</import-only-module>
</module-set>
<schema>
  <name>test-schema</name>
  <module-set>mountee-set</module-set>
</schema>
<datastore>
  <name>ds:running</name>
  <schema>test-schema</schema>
</datastore>
<datastore>
  <name>ds:operational</name>
  <schema>test-schema</schema>
</datastore>
<content-id>2</content-id>
</yang-library>
<modules-state xmlns="urn:ietf:params:xml:ns:yang:ietf-yang-library">
  <module-set-id>2</module-set-id>
</modules-state>
<schema-mounts
  xmlns="urn:ietf:params:xml:ns:yang:ietf-yang-schema-mount">
  <mount-point>
    <module>example-collection-manifest</module>
```

```
<label>yang-push-collection</label>
<shared-schema/>
</mount-point>
</schema-mounts>
<CODE ENDS>

<CODE BEGINS> file "data-collection-toplevel-yanglib.xml"
<yang-library xmlns="urn:ietf:params:xml:ns:yang:ietf-yang-library"
  xmlns:ds="urn:ietf:params:xml:ns:yang:ietf-datastores">
  <module-set>
    <name>main-set</name>
    <module>
      <name>ietf-datastores</name>
      <revision>2018-02-14</revision>
      <namespace>
        urn:ietf:params:xml:ns:yang:ietf-datastores
      </namespace>
    </module>
    <module>
      <name>ietf-yang-library</name>
      <revision>2019-01-04</revision>
      <namespace>
        urn:ietf:params:xml:ns:yang:ietf-yang-library
      </namespace>
    </module>
    <module>
      <name>ietf-yang-schema-mount</name>
      <revision>2019-01-14</revision>
      <namespace>
        urn:ietf:params:xml:ns:yang:ietf-yang-schema-mount
      </namespace>
    </module>
    <module>
      <name>example-collection-manifest</name>
      <revision>2025-02-21</revision>
      <namespace>
        http://example.org/example-collection-manifest
      </namespace>
    </module>
    <module>
      <name>ietf-platform-manifest</name>
      <revision>2025-02-21</revision>
      <namespace>
        urn:ietf:params:xml:ns:yang:ietf-platform-manifest
      </namespace>
    </module>
    <import-only-module>
      <name>ietf-inet-types</name>
```



```
<revision>2013-07-15</revision>
<namespace>
  urn:ietf:params:xml:ns:yang:ietf-inet-types
</namespace>
</import-only-module>
<import-only-module>
  <name>ietf-yang-types</name>
  <revision>2023-01-23</revision>
  <namespace>
    urn:ietf:params:xml:ns:yang:ietf-yang-types
  </namespace>
</import-only-module>
</module-set>
<schema>
  <name>main-schema</name>
  <module-set>main-set</module-set>
</schema>
<datastore>
  <name>ds:running</name>
  <schema>main-schema</schema>
</datastore>
<datastore>
  <name>ds:operational</name>
  <schema>main-schema</schema>
</datastore>
<content-id>1</content-id>
</yang-library>
<modules-state
  xmlns="urn:ietf:params:xml:ns:yang:ietf-yang-library">
  <module-set-id>2</module-set-id>
</modules-state>
<CODE ENDS>
```

Appendix D. Validating the Example

This section provides the command line for validating the example in Figure 4 using [yanglint]. The files "data-collection-extension-data.xml" and "data-collection-toplevel-yanglib.xml" are provided in the previous section. The file "manifests-example.json" is the one from Figure 4.

```
yanglint -e -x data-collection-extension-data.xml \
  -Y data-collection-toplevel-yanglib.xml \
  manifests-example.json
```

Acknowledgements

Thanks to Mohamed Boucadair, Tianran Zhou, Jan Lindblad, Ahmed Elhassany, Joe Clarke, Alex Huang Fang, Zhuoyao Lin and Quifang Ma for their reviews and comments.

Authors' Addresses

Benoit Claise
Everything OPS
Email: benoit@everything-ops.net

Jean Quilbeuf
Huawei
Email: jean.quilbeuf@huawei.com

Diego R. Lopez
Telefonica I+D
Don Ramon de la Cruz, 82
Madrid 28006
Spain
Email: diego.r.lopez@telefonica.com

Ignacio Dominguez
Telefonica I+D
Ronda de la Comunicacion, S/N
Madrid 28050
Spain
Email: ignacio.dominguezmartinez@telefonica.com

Thomas Graf
Swisscom
Binzring 17
CH-8045 Zurich
Switzerland
Email: thomas.graf@swisscom.com