YangPush Notification Capabilities
draft-ietf-netconf-notification-capabilities-01

Abstract

This document proposes a YANG module that allows a YANG server to specify server capabilities related to "Subscription to YANG Datastores" (YangPush). It proposes to use YANG Instance Data to document this information already in implementation time.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on September 2, 2019.

Copyright Notice

Copyright (c) 2019 IETF Trust and the persons identified as the document authors. All rights reserved.

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

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

The terms Yang-Push, On-change subscription and Periodic subscription are used as defined in [I-D.ietf-netconf-yang-push]

On-change Notification Capability: The capability of the YANG server to support On-change subscriptions.

Implementation-time information: Information about the YANG server’s behavior that is made available during the implementation of the server, available from a source other than a running Yang server.

Runtime-information: Information about the YANG server’s behavior that is available from the running YANG server via a protocol like NETCONF, RESTCONF or HTTPS.

2. Introduction

As defined in [I-D.ietf-netconf-yang-push] a YANG server may allow clients to subscribe to updates from a datastore and subsequently push such update notifications to the client. Notifications may be sent periodically or on-change (more or less immediately after each change).
A YANG server supporting YANG-Push has a number of capabilities that are determined during the implementation of the server. These include:

- Supported dampening periods for on-change subscriptions
- Supported (reporting) periods for periodic subscriptions
- Maximum number of objects that can be sent in an update
- The set of data nodes for which on-change notification is supported

Servers MAY have limitations in how many update notifications and how many datastore node updates they can send out in a certain time-period.

In some cases, a publisher supporting on-change notifications will not be able to push updates for some object types on-change. Reasons for this might be that the value of the datastore node changes frequently (e.g., in-octets counter), that small object changes are frequent and meaningless (e.g., a temperature gauge changing 0.1 degrees), or that the implementation is not capable of on-change notification for a particular object. In those cases, it will be important for client applications to have a way to identify for which objects on-change notifications are supported and for which ones not.

Faced with the reality that support for on-change notification does not mean that such notifications will be sent for any specific data node, client/management applications can not rely on the on-change functionality unless the client has some means to identify for which objects on-change notifications are supported. YANG models are meant to be used as an interface contract. Without identification of data nodes supporting on-change, this contract would only state the YANG server may (or may not) send on-change notifications for a data node specified in a YANG module.

This document proposes a YANG module that allows a client to discover YANG-Push related capabilities.

YANG-Push related capability information will be needed both in implementation-time and run-time.

Implementation time information is needed by Network Management System (NMS) implementers. During NMS implementation for any functionality that depends on the notifications the information about on-change notification capability is needed. If the information is not available early in some document, but only as instance data from
the network node, the NMS implementation will be delayed, because it has to wait for the network node to be ready. Also assuming that all NMS implementers will have a correctly configured network node available to retrieve data from, is an expensive proposition. (An NMS may handle dozens of network node types.) Often a fully functional NMS is a requirement for introducing a new network node type into a network, so delaying the NMS effectively delays the availability of the network node as well.

Implementation time information is needed by system integrators. When introducing a network node type into their network, operators often need to integrate the node type into their own management system. The NMS may have management functions that depend on on-change notifications. The network operator needs to plan his management practices and NMS implementation before he even decides to buy the specific network node type. Moreover the decision to buy the node type sometimes depends on these management possibilities.

Run-time information is needed

- for any "purely model driven" client, e.g. a NETCONF-browser. As long as it has a valid model to read the capability information, it does not care which data nodes send notification, it will just handle what is available.

- in case the capability might change during run-time e.g. due to licensing, HW constraints etc.

- to check that early, implementation time capability information about the capabilities is indeed what the server implements (is the supplied documentation correct?)

3. Notification Capability Model

It is a goal to provide YangPush notification capability information in a format that is

- vendor independent (standard)

- formal (no freeform English text please)

- the same both in implementation-time and run-time

The YANG module ietf-notification-capabilities is defined to provide the information. It contains

- a set of capabilities related to the amount of notifications the server can send out
a default on-change notification capability separately for config false and config true data nodes

an on-change-notification-capability list containing a potentially different true/false notification capability for a few data nodes in the schema tree. Unless a node is in this list with a specific capability value, it inherits its on-change-notification-capability from its parent in the data tree, or from the relevant default values. It is assumed that only a small number of nodes will be included in this list: special cases where the default behavior is not followed. For a detailed description of the usage of this list see the description in the YANG module.

The information SHALL be provided in two ways both following the ietf-notification-capabilities module:

- It SHALL be provided by the implementer as YANG instance data file complying to the [I-D.lengyel-netmod-yang-instance-data]. The file SHALL be available already in implementation time retrievable in a way that does not depend on a live network node. E.g. download from product Website.

- It SHALL be available via NETCONF or RESTCONF from the live YANG server during runtime.

3.1. Tree Diagram

The following tree diagram [RFC8340] provides an overview of the data model.

```plaintext
module: ietf-notification-capabilities
  +--ro yangpush-notification-capabilities
    +--ro minimum-dampening-period? uint32
    +--ro (update-period)?
      |  +--:(minimum-update-period)
      |  |  +--ro minimum-update-period? uint32
      |  +--:(supported-update-period)
      |     +--ro supported-update-period* uint32
    +--ro max-objects-per-update? uint32
    +--ro notification-sent-for-config-default? boolean
    +--ro notification-sent-for-state-default? boolean
    +--ro on-change-notification-capability* [node-selector]
      +--ro node-selector nacm:node-instance-identifier
      +--ro on-change-notification-sent boolean
```
3.2. YANG Module

```yang
module ietf-notification-capabilities {
  yang-version 1.1;
  namespace
    "urn:ietf:params:xml:ns:yang:ietf-notification-capabilities";
  prefix inc;

  import ietf-netconf-acm { prefix nacm; }

  organization
    "IETF NETCONF (Network Configuration) Working Group";

  contact
    "WG Web: <https://datatracker.ietf.org/wg/netconf/>
    WG List: <mailto:netconf@ietf.org>
    WG Chair: Kent Watsen
      <mailto:kwatsen@juniper.net>
    WG Chair: Mahesh Jethanandani
      <mailto:mjethanandani@gmail.com>
    Editor: Balazs Lengyel
      <mailto:balazs.lengyel@ericsson.com>"

  description "This module specifies YANG-Push related server
  capabilities. It contains
  - capabilities related to the amount of notifications the
    server can send out
  - default and schema node specific information specifying
    the set of data nodes for which the YANG server is capable
    of sending on-change notifications.

  On-change notification capability is marked as true or false.
  This marking is inherited from the parent down the data tree
  unless explicitly marked otherwise.

  On-change notifications SHALL be sent for a config=true
  data node if one of the following is true:
  - if it is a top level data-node and is not specified in the
    on-change-notification-capability list and the
    notification-sent-for-config-default is true; or
  - notifications are sent for its parent data node and it is
    not specified in the on-change-notification-capability list; or
```
- it is specified in the on-change-notification-capability list and has a on-change-notification-sent value true.

On-change notifications SHALL be sent for a config=false data node if one of the following is true:
- if it is a top level data-node or has a config=true parent data node and is not specified in the on-change-notification-capability list and the notification-sent-for-state-default is true; or
- notifications are sent for its parent data node which is also config=false and it is not specified in the on-change-notification-capability list; or
- it is specified in the on-change-notification-capability list and has an on-change-notification-sent value true or

revision 2019-02-28 {
  description "Initial version";
  reference
    "RFC XXX: YangPush Notification Capabilities"
}

container yangpush-notification-capabilities {
  config false;
  description "YANG-Push related server capabilities";

  leaf minimum-dampening-period {
    type uint32;
    units msec;
    description "The minimum dampening period supported for on-change subscriptions.";
  }

  choice update-period {
    description "Supported period values.";
    leaf minimum-update-period {
      type uint32;
      units centiseconds;
      description "Minimum update period supported for a periodic subscription.";
    }
  }

  leaf-list supported-update-period {
    type uint32;
    units centiseconds;
    description "Specific supported update period values for a periodic subscription";
  }

leaf max-objects-per-update {
  type uint32;
  description "Maximum number of objects that can be sent in an update";
}

leaf notification-sent-for-config-default {
  type boolean;
  default true;
  description "Specifies the default value for top level configuration data nodes for the on-change-notification-sent capability.";
}

leaf notification-sent-for-state-default {
  type boolean;
  default false;
  description "Specifies the default value top level state data nodes for the on-change-notification-sent capability.";
}

list on-change-notification-capability {
  key node-selector;
  description "A list of data nodes that have the on-change-notification-capability specifically defined. Should be used when specific data nodes support on-change notification in a module/subtree that generally does not support it or when some data nodes do not support the notification in a module/subtree that generally supports on-change notifications.";

  leaf node-selector {
    type nacm:node-instance-identifier;
    description "Selects the data nodes for which on-change capability is specified.";
  }

  leaf on-change-notification-sent {
    type boolean;
    mandatory true;
    description "Specifies whether the YANG server will send on-change notifications for the selected data nodes.";
  }
}
4. Security Considerations

The YANG module defined in this document is designed to be accessed via YANG based management protocols, such as NETCONF and RESTCONF. Both of these protocols have mandatory-to-implement secure transport layers (e.g., SSH, TLS) with mutual authentication.

The NETCONF access control model (NACM) provides the means to restrict access for particular users to a pre-configured subset of all available protocol operations and content.

The data in this module is not security sensitive.

5. IANA Considerations

5.1. The IETF XML Registry

This document registers one URI in the IETF XML registry [RFC3688]. Following the format in [RFC3688], the following registrations are requested:

Registrant Contact: The NETCONF WG of the IETF.
XML: N/A, the requested URI is an XML namespace.

5.2. The YANG Module Names Registry

This document registers one YANG module in the YANG Module Names registry [RFC7950]. Following the format in [RFC7950], the the following registrations are requested:

name: ietf-notification-capabilities
prefix: inc
reference: RFC XXXX

6. Open Issues

Do we need separate defaults/individual lists for every datastore? Proposal: no, it would be an overkill.
Internet-Draft     YangPush Notification Capabilities         March 2019

Should type nacm:node-instance-identifier be moved to yang-types?
It is useful for more than just nacm.

7. References

7.1. Normative References

[I-D.ietf-netconf-yang-push]

[I-D.lengyel-netmod-yang-instance-data]


7.2. Informative References


Appendix A. Changes between revisions

v00 - v01

- Add more capabilities: minimum period, supported period max-number of objects, min dampening period, dampening supported
Authors’ Addresses

Balazs Lengyel
Ericsson
Magyar Tudosok korutja 11
1117 Budapest
Hungary

Email: balazs.lengyel@ericsson.com

Alexander Clemm
Huawei USA
2330 Central Expressway
Santa Clara, CA 95050
USA

Email: ludwig@clemm.org