                   YangPush Notification Capabilities
              draft-lengyel-netconf-notification-capabilities-02

Abstract

   This document proposes a YANG module that allows a YANG server to
   specify for which data nodes it will send "YANG Datastore
   Subscription" on-change notifications. It also proposes to use YANG
   Instance Data to document this information 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 January 3, 2019.

Table of Contents

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.

   On-change Notification Capability: The capability of the YANG server
   to send on-change notifications on the change of the value for a
   specific data node.

   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 then a running Yang server.

   Rutime-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).

   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 are
not supported.

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 and for which ones are
not 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 identify
which data nodes support on-change notification, removing the
uncertainty for on-change notifications.

On-change Notification Capability information will be needed both in
implementation-time and run-time.

Run-time information is needed

o for any "purely model driven" client, e.g. a Netconf-browser. As
   long as it has a valid model, it does not care which data nodes
   send notification, it will just handle whats available.

o to check that early implementation time information about the
   capability is indeed what the server supports

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

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.
System integrators will need information about on change notification
capability early. 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.

3. On-change Notification Capability Model

As described above a number of stakeholders need information about
the on change notification capability both in implementation and run-
time. It is a goal to provide this information in a format that is

o vendor independent (standard)

o formal (no freeform English text please)

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

The YANG module ietf-notification-capabilities is defined to provide
information about the on-change notification capabilities. There is
a default notification capability separately for config false and
config true data nodes. There is also an on-change-notification-
capability list containing a potentially different true/false
notification capability for any data node in the schema tree. Unless
a node is in the 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.

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

o 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.

o 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.

```
 module: ietf-notification-capabilities
     +--ro on-change-notification-capability
        +--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

   <CODE BEGINS> file "ietf-notification-capabilities.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 for which data nodes will the
    YANG server send 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:
- it is specified in the on-change-notification-capability
list and has a on-change-notification-sent value true or
- notifications are sent for its parent data node and it is
not specified in the on-change-notification-capability list or
- 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.

On-change notifications SHALL be sent for a config=false
data node if one of the following is true:
- it is specified in the on-change-notification-capability
list and has an on-change-notification-sent value 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
- 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.
";

reference "RFC XXXX Yang-Push";

```
  revision 2018-07-02 {
    description "Initial version";
    reference
      "RFC XXX: YangPush Notification Capabilities";
  }

  container on-change-notification-capability {
    config false;
    description "Contains default values for the
      on-change-notifiction-capability and a list of data nodes that
      have the on-change-notification-capability specifically defined.";

    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;
     }

     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.";
     }
   }
  }
}

   <CODE ENDS>
```

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.

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:

      URI: urn:ietf:params:xml:ns:yang:ietf-notification-capabilities
      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
   namespace: urn:ietf:params:xml:ns:yang: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.

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

7. References

7.1. Normative References

   [I-D.ietf-netconf-yang-push]
              Clemm, A., Voit, E., Prieto, A., Tripathy, A., Nilsen-
              Nygaard, E., Bierman, A., and B. Lengyel, "YANG Datastore
              Subscription", draft-ietf-netconf-yang-push-17 (work in
              progress), July 2018.

   [I-D.lengyel-netmod-yang-instance-data]
              Lengyel, B. and B. Claise, "YANG Instance Data Files and
              their use for Documenting Server Capabilities", draft-
              lengyel-netmod-yang-instance-data-02 (work in progress),
              July 2018.

   [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>.

7.2. Informative References

   [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate
              Requirement Levels", BCP 14, RFC 2119,
              DOI 10.17487/RFC2119, March 1997,
              <https://www.rfc-editor.org/info/rfc2119>.

   [RFC3688] Mealling, M., "The IETF XML Registry", BCP 81, RFC 3688,
              DOI 10.17487/RFC3688, January 2004,
              <https://www.rfc-editor.org/info/rfc3688>.

   [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>.

   [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. Changes between revisions

   v01 - v02

   o Instead of augmenting ietf-yang-library a more simple standalone
      model is proposed.

   v00 - v01

   o Corrections

   o Augment only the new yanglib

   o Correct the condtions for notifying state data

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