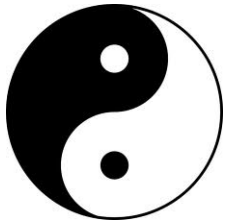

YANG-PUSH ON-CHANGE NOTIFICATION CAPABILITY

Balazs Lengyel

Alex Clemm

draft-lengyel-netconf-notification-capabilities

2018-07-15

A CONTRACT – "MAYBE"

- Publishers supporting on-change notifications will not always be able to push on-change updates for every object type.
  - Not implemented because
    - Too frequent change (inOctets counter)
      - (Note: SmartFilters may provide a solution -> later)
    - Meaningless small change (temperature changing 0.1 degrees)
    - Any other reason
  - Resource limitation (missing HW)
  - Small constrained network node using a common model with Big systems

- Support for on-change does not mean that notifications are sent for any specific object

YANG MODEL BASED

- Lets document on-change capabilities. But how:
  - vendor independent (standard)
  - formal (no free form English text please)
  - Information needed both during
    - implementation-time - for NMS developers, system integrators
    - run-time - useful especially if the capability might change
  - Same format both in implementation-time and run-time
- So let's make it a YANG Module
  - Describe for each data node whether it supports on-change notification (with default "not supported unless specified")
- Instantiated by server during runtime
- YANG Instance Data for implementation-time
  - draft-netmod-lengyel-yang-instance-data

IETF-NOTIFICATION-CAPABILITIES.YANG

- Simplified YANG module
- Standalone model
  - One model to cover full YANG server model, not list entry per YANG module implemented
  - Not augmenting ietf-yang-library
  - Not augmenting ietf-subscribed-notifications as it has no suitable root container
- Default values per YANG server – state/config
- Data node specific values
- Effective capability value inherited down the data tree
- Due to defaults and inheritance only few specific markings needed

IETF-NOTIFICATION-CAPABILITIES.YANG

```
module: ietf-notification-capabilities
    +--ro on-change-notification-capability
       +--ro config-default?              boolean
       +--ro state-default?               boolean
       +--ro notification-capability* [node-selector]
          +--ro node-selector             nacm:node-instance-identifier
          +--ro notification-sent         boolean
```

IETF-NOTIFICATION-CAPABILITIES.YANG

```
<instance-data-set xmlns=
  "urn:ietf:params:xml:ns:yang:ietf-yang-instance-data“>
  <name>acme-router-on-change-capability</name>
  <revision>2108-01-25</revision>
  <description>Notification capability </description>
  <data>
    <on-change-notification-capability
      xmlns="urn:ietf:params:xml:ns:yang:ietf-notification-capabilities">
      <state-default>true</state-default>
      <notification-capability>
        <node-selector
          xmlns:sys=urn:ietf:params:xml:ns:yang:ietf-system>
            /sys:system-state/sys:clock/sys:current-datetime
        </node-selector>
        <notification-sent>false</notification-sent>
      </notification-capability>
    </on-change-notification-capability>
  </data>
</instance-data>
```

OPEN ISSUES – WAY FORWARD

- Model capabilities separately for each NMDA datastore?
  - Proposal: No, it would be an overkill
- Names in model too long
  - Module structure should be OK
  - Make names shorter

- Request adoption as workgroup document

- Received support on last IETF
- Received support and comments on the mailing list