

# **YANG model for NETCONF Event Notifications**

draft-ahuang-netconf-notif-yang-04

**A. Huang Feng**, INSA-Lyon  
P. Francois, INSA-Lyon  
T. Graf, Swisscom  
B. Claise, Huawei

March 19th 2023

# YANG model for NETCONF Event Notifications

## Context - Netconf Notification

```
<notification xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0">
  <eventTime>2022-09-02T10:59:55.32Z</eventTime>
  <push-update xmlns="urn:ietf:params:xml:ns:yang:ietf-yang-push">
    <id>101</id>
    <datastore-contents>
      <interfaces xmlns="urn:ietf:params:xml:ns:yang:ietf-interfaces">
        <interface>
          <name>eth0</name>
          <oper-status>up</oper-status>
        </interface>
      </interfaces>
    </datastore-contents>
  </push-update>
</notification>
```

```
{
  "ietf-notification:notification": {
    "eventTime": "2017-10-25T08:00:11.22Z",
    "ietf-yang-push:push-update": {
      "id": 1011,
      "datastore-contents": {
        "ietf-interfaces:interfaces": [
          "interface": {
            "name": "eth0",
            "oper-status": "up"
          }
        ]
      }
    }
  }
}
```

RFC 5277 - Netconf Event Notifications

RFC 8641 - YANG Push

YANG encodings:

- RFC 7950 - YANG XML
- RFC 7951 - YANG JSON
- RFC 9254 - YANG CBOR

# YANG model for NETCONF Event Notifications

## Proposal

- Definition of the notification structure in a YANG
- Updates RFC5277 (NETCONF Notifications)
- Uses the same XML URI as RFC5277
- “eventTime” in CamelCase following model defined in RFC5277

```

sx:structure notification {
  leaf eventTime {
    type yang:date-and-time;
    mandatory true;
    description
      "The date and time the event was generated by the event source.
      This parameter is of type dateTime and compliant to [RFC3339].
      Implementations must support time zones.
      The leaf name in camel case matches the name of the XSD element
      defined in Section 4 of RFC5277.";
  }
}

```

```

module: ietf-notification

  structure notification:
    +-- eventTime      yang:date-and-time

```

# **YANG model for NETCONF Event Notifications**

## Current status

- Asks for SID Range for YANG-CBOR using YANG-SID (draft-ietf-core-sid)
- Added XML example
- Propose SID file in the appendix

# YANG model for NETCONF Event Notifications

## Next steps

- Request more feedback from the WG
- **Working Group Adoption?**
- Inconsistencies in the Notification definition in YANG-JSON and YANG-CBOR
  - “Notification” well defined in RFC9750 (YANG 1.1)
  - “Notifications” not explicitly defined in RFC7951 (YANG-JSON) and RFC9254 (YANG-CBOR)
  - issue raised in *draft-netana-nmop-yang-kafka-integration-00*

## BACK UP: Difference with *draft-ietf-netconf-notification-messages*

### draft-ahuang-netconf-notif-yang

```
module: ietf-notification  
  
  structure notification:  
    +-- eventTime    yang:date-and-time
```

### draft-ietf-netconf-notification-messages

```
structure message  
+--ro message!  
  +--ro message-header  
  | +--ro message-time          yang:date-and-time  
  | +--ro message-id?          uint32  
  | +--ro message-generator-id? string  
  | +--ro notification-count?   uint16  
  +--ro notifications*  
  | +--ro notification-header  
  | | +--ro notification-time      yang:date-and-time  
  | | +--ro yang-module?          yang:yang-identifier  
  | | +--ro subscription-id*       uint32  
  | | +--ro notification-id?       uint32  
  | | +--ro observation-domain-id? string  
  | +--ro notification-contents?  
  | +--ro notification-footer!  
  | | +--ro signature-algorithm    string  
  | | +--ro signature-value        string  
  | | +--ro integrity-evidence?    string  
  +--ro message-footer!  
  | +--ro signature-algorithm      string  
  | +--ro signature-value          string  
  | +--ro integrity-evidence?      string
```

## BACK UP 2: Notifications encoded in XML/YANG-JSON/YANG-CBOR

- RFC5277 (NETCONF Event Notifications) Defines the structure of the Notification and XML examples:

```
<notification
  xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0">
  <eventTime>2007-07-08T00:02:00Z</eventTime>
  <event xmlns="http://example.com/event/1.0">
    <eventClass>fault</eventClass>
    <reportingEntity>
      <card>Ethernet2</card>
    </reportingEntity>
    <severity>critical</severity>
  </event>
</notification>
```

- RFC7950 (YANG 1.1) Section 4.2.10, defines how Notifications should be encoded when modeled in YANG:

YANG Example:

```
notification link-failure {
  description
    "A link failure has been detected.";
  leaf if-name {
    type leafref {
      path "/interface/name";
    }
  }
  leaf if-admin-status {
    type admin-status;
  }
  leaf if-oper-status {
    type oper-status;
  }
}
```

NETCONF XML Example:

```
<notification
  xmlns="urn:ietf:params:netconf:capability:notification:1.0">
  <eventTime>2007-09-01T10:00:00Z</eventTime>
  <link-failure xmlns="urn:example:system">
    <if-name>so-1/2/3.0</if-name>
    <if-admin-status>up</if-admin-status>
    <if-oper-status>down</if-oper-status>
  </link-failure>
</notification>
```

## BACK UP 2: Notifications encoded in XML/YANG-JSON/YANG-CBOR

- RFC7951 (YANG-JSON):
  - Notifications are **not explicitly** covered
  - Example in Section 5.5 covering how “anydata” statements should be encoded:

Example: For the anydata definition

anydata data;

the following is a valid JSON-encoded instance:

```
"data": {  
  "ietf-notification:notification": {  
    "eventTime": "2014-07-29T13:43:01Z",  
    "example-event:event": {  
      "event-class": "fault",  
      "reporting-entity": {  
        "card": "Ethernet0"  
      },  
      "severity": "major"  
    },  
  },  
}
```

*eventTime* as defined in RFC5277 present





# BACK UP 2: Notifications encoded in XML/YANG-JSON/YANG-CBOR

- RFC9254 (YANG-CBOR):
  - Notifications are **not explicitly** covered
  - Defines a Notification as a “container-like” structure:

## container-like instance:

An instance of a container, a YANG data structure, notification contents, RPC input, RPC output, action input, or action output ([Section 4.2](#)); a list entry in a list ([Section 4.4](#)); or an anydata node ([Section 4.5](#)).

- An example of a “notification” statement is in Section 4.5 covering “anydata” statements

```
module example-port {  
  ...  
  
  notification example-port-fault { // SID 60200  
    leaf port-name { // SID 60201  
      type string;  
    }  
    leaf port-fault { // SID 60202  
      type string;  
    }  
  }  
}
```

```
{  
  60123 : { // last-event (SID 60123) /  
    47(60200) : { // event-port-fault (SID 60200) /  
      1 : "0/4/21", // port-name (SID 60201) /  
      2 : "Open pin 2" // port-fault (SID 60202) /  
    }  
  }  
}
```

## Using YANG-SID

```
{  
  "event-log:last-event" : {  
    "example-port:example-port-fault" : {  
      "port-name" : "0/4/21",  
      "port-fault" : "Open pin 2"  
    }  
  }  
}
```

*ietf-notification:notification* container missing  
*eventTime* leaf missing

## Using names in keys