A YANG Data model for Policy based Event Management
draft-wwx-netmod-event-yang-01

Abstract

[RFC8328] defines a policy-based management framework that allow
definition of a data model to be used to represent high-level,
possibly network-wide policies. This document defines an YANG data
model for the policy based event management [RFC7950]. The policy
based Event YANG provides the ability for the network management
function (within a controller, an orchestrator, or a network element)
to control the configuration and monitor state change on the network
element and take simple and instant action when a trigger condition
on the system state is met.

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
1. Introduction

[RFC8328] defines a policy-based management framework that allows definition of a data model to be used to represent high-level, possibly network-wide policies. This document defines an policy based Event Management YANG data model [RFC7950]. The policy based Event management YANG provides the ability for the network management function (within a controller, an orchestrator, or a network element) to monitor state changes on the network element and take simple and instant action when a trigger condition on the system state is met.

The data model in this document is designed to be compliant with the Network Management Datastore Architecture (NMDA) [RFC8342].

2. Conventions used in this document

2.1. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119]. In this document, these words will appear with that interpretation only when
in ALL CAPS. Lower case uses of these words are not to be interpreted as carrying [RFC2119] significance.

This document uses the following terms:

Error  A deviation of a system from normal operation [RFC3877].

Fault  Lasting error or warning condition [RFC3877].

Event  Something that happens which may be of interest or trigger the invocation of the rule. A fault, an alarm, a change in network state, network security threat, hardware malfunction, buffer utilization crossing a threshold, network connection setup, an external input to the system, for example [RFC3877].

2.2. Tree Diagrams

Tree diagrams used in this document follow the notation defined in [RFC8340].

3. Objectives

This section describes some of the design objectives for the policy based Event management Data Model:

- The policy based Event management YANG should provide the ability for the network management function to control configuration and monitor state changes on a network element using the NETCONF/RESTCONF, and initiate simple actions whenever a trigger condition is met. For example, a NETCONF subscribed notification can be generated when a system state value exceeds the threshold.

- Clear and precise identification of policy based Event types and managed objects.

- Allow the server to inform the client that certain Event are related to other Event.

- Allow one event to be able to trigger another external event or generate derived events.

- The event data model defined in this document can be implemented on the management system that also implements EVENT-MIB; thus, the mapping between the event data model and ENTITY-MIB should be clear.
4. Relationship to YANG Push

YANG-push mechanism provides a subscription service for updates from a datastore. And it supports two types of subscriptions which are distinguished by how updates are triggered: periodic and on-change.

The On-change Push allow receivers to receive updates whenever changes to target managed objects occur. This document specifies a mechanism that provides three trigger conditions:

- **Existence**: When a specific managed object appears, the trigger fires, e.g. reserved ports are configured.

- **Boolean**: The user can set the type of boolean comparison (e.g. unequal, equal, less, less-or-equal, greater, greater-or-equal, etc). If the test result is true, the trigger fires. The trigger will not fire again until the test result has become false and fall back to be true. e.g., when the boolean comparison type is ‘less’, the trigger will be fired if the system state value of managed object is less than the value set for the target managed object.

- **Threshold**: The event that may be triggered when a managed object in multiple instances of the data tree is found and exceed the pre-configured threshold. For example, when the system state value of that managed object is greater than or equal to ‘rising-value’ and the ‘startup’ is equal to ‘rising’, then one threshold rising event is triggered for that managed object. Another example is if the system state value of the managed object is less than or equal to ‘falling-value’ and the ‘startup’ is equal to ‘falling’ then one threshold falling event is triggered for that managed object.

And the YANG Push mechanism more focus on the remote mirroring and monitoring of configuration and operational state. For example, for on change method, the subscriber will receive a notification if the changes occurs. The model defined in this document provides a method which allow automatic adjusting the value of the corresponding managed object when some event is triggered. It establishes connection between network service monitoring and network service provision and can use output generated by network service monitoring as input of network service provision and thereby provide automated network management. The details of the usage example is described in Appendix A.
5. Relationship to EVENT MIB

If the device implements the EVENT-MIB [RFC2981], each entry in the "/events/event/trigger" list is mapped to MteTriggerEntry, MteTriggerExistenceEntry, MteTriggerBooleanEntry, MteTriggerThresholdEntry, MteObjectsEntry, MteEventEntry, MteEventSetEntry, respectively.

The following table lists the YANG data nodes with corresponding objects in the EVENT-MIB [RFC2981].
### YANG data node in

<table>
<thead>
<tr>
<th>ietf-event.yang</th>
<th>EVENT-MIB Objects</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>(RFC2981)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>min-data-object</th>
<th>mteResourceSampleMinimum</th>
</tr>
</thead>
<tbody>
<tr>
<td>max-data-object</td>
<td>mteResourceSampleInstanceMaximum</td>
</tr>
<tr>
<td>target</td>
<td>mteObjectsName</td>
</tr>
<tr>
<td>event-name</td>
<td>mteEventName</td>
</tr>
<tr>
<td>event-description</td>
<td>mteEventComment</td>
</tr>
<tr>
<td>value</td>
<td>mteEventSetValue</td>
</tr>
<tr>
<td>events/event/trigger/name</td>
<td>mteTriggerName</td>
</tr>
<tr>
<td>trigger-description</td>
<td>mteTriggerComment</td>
</tr>
<tr>
<td>frequency</td>
<td>mteTriggerFrequency</td>
</tr>
<tr>
<td>comparison</td>
<td>mteTriggerBooleanComparison</td>
</tr>
<tr>
<td>value</td>
<td>mteTriggerBooleanValue</td>
</tr>
<tr>
<td>rising-event</td>
<td>mteTriggerThresholdRising</td>
</tr>
<tr>
<td>falling-event</td>
<td>mteTriggerThresholdFalling</td>
</tr>
<tr>
<td>delta-rising-event</td>
<td>mteTriggerThresholdDeltaRising</td>
</tr>
<tr>
<td>threshold/startup</td>
<td>mteTriggerThresholdStartup</td>
</tr>
<tr>
<td>existence/enable</td>
<td>mteTriggerExistenceStartup</td>
</tr>
<tr>
<td>boolean/enable</td>
<td>mteTriggerBooleanStartup</td>
</tr>
</tbody>
</table>

---

### 6. Model Overview

The YANG data model for the Event management has been split into two modules:
The ietf-event-trigger.yang module defines a grouping for a generic trigger. It is intended that this grouping will be used by the policy based event management model or other models that require the trigger conditions. In this model, three trigger conditions are defined under the "test" choice node:

* Existence: When a specific managed object appears, the trigger fires.

* Boolean: If the test result on specific managed object is true the trigger fires. The Boolean trigger condition is used when threshold value is a static value or pre-configured value.

* Threshold: The event that may be triggered when the system state value of a managed object within data-instance list exceed the threshold. The threshold trigger condition is used when threshold value changes over time or the system state value of managed object changes in ascend or descend order.

The ietf-event.yang module defines four lists: trigger, target, event, and action. Triggers define the targets meeting some conditions that lead to events. Events trigger corresponding actions:

* Each trigger can be seen as a logical test that, if satisfied or evaluated to be true, cause the action to be carried out. The ietf-event.yang module uses groupings defined in ietf-event-trigger.yang to present the trigger attributes.

* The target list defines managed objects that can be added to notifications or be set to a new value on the trigger, the trigger test type, or the event that resulted in the actions.

* The event list defines what happens when an event is triggered, i.e., trigger corresponding action, e.g., sending a notification, setting a value to the managed object or both.

* The action list consists of updates or invocations on local managed object attributes and defines a set of actions which will be performed (e.g. notification, set, another event, etc) when corresponding event be triggered. The value to be set can use many variations on rule structure.

The following tree diagrams [RFC8340] provide an overview of the data model for "ietf-event-trigger" module and the "ietf-event" module.

```
trigger-grouping
```

module: ietf-event

++--rw events
  +--rw min-data-object?  uint32
  +--rw max-data-object?  uint32
  +--rw event* [event-name type]
    +--rw event-name  string
    +--rw type  identityref
    +--rw event-description?  trig:target
    +--rw clear?  boolean
    +--rw related-event* [event-name type]
      | +--rw event-name  string
      | +--rw type  identityref
    +--rw trigger* [name]
      | +--rw name  string
      | +--rw type?  enumeration
      | +--rw trigger-description?  string
      | +--rw frequency
        | +--rw type?  identityref
        | +--rw periodic
          | +--rw interval  uint32
          | +--rw start?  yang:date-and-time
          | +--rw end?  yang:date-and-time
          | +--rw scheduling
            | +--rw month*  string
            | +--rw day-of-month*  uint8
The relation between Event, Trigger, Target and Action is described as follows:
One event may trigger another event, i.e., the action output in the first event can be input to target in the second event, but if it does not trigger another event, the relation between action and target should be ignored.

7. EVENT TRIGGER YANG Module

<CODE BEGINS> file "ietf-event-trigger@2018-12-18.yang"
module ietf-event-trigger {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-event-trigger";
  prefix trig;

  import ietf-yang-types {
    prefix yang;
  }

  organization
    "IETF xxx Working Group";
  contact
    "Zitao Wang: wangzitao@huawei.com
    Qin Wu: bill.wu@huawei.com";
  description
    "This module defines a reusable grouping for event trigger."

  revision 2018-12-18 {
    description
      "Initial revision."
      reference "foo";
  }

  typedef match-value {
    type union {
      
    }
}
typedef target {
    type union {
        type instance-identifier;
        type yang:object-identifier;
        type yang:uuid;
        type string;
    }
    description
    "If the target is modelled in YANG, this type will be an instance-identifier. If the target is an SNMP object, the type will be an object-identifier. If the target is anything else, for example a distinguished name or a CIM path, this type will be a string. If the target is identified by a UUID use the uuid type. If the server supports several models, the precedence should be in the order as given in the union definition.";
}

grouping trigger-grouping {
    description
    "A grouping that provides event trigger.";
    choice test {
        description
        "Choice test";
        container existences {
            leaf-list target {
                type target;
                description
                "List for target objects";
            }
            description
            "Container for existence";
        }
        container boolean {
            leaf comparison {
                type boolean;
                description
                "Comparison for boolean";
            }
            description
            "Container for boolean";
        }
    }
}
type enumeration {
    enum "unequal" {
        description "Indicates that the comparison type is unequal to.";
    }
    enum "equal" {
        description "Indicates that the comparison type is equal to.";
    }
    enum "less" {
        description "Indicates that the comparison type is less than.";
    }
    enum "less-or-equal" {
        description "Indicates that the comparison type is less than or equal to.";
    }
    enum "greater" {
        description "Indicates that the comparison type is greater than.";
    }
    enum "greater-or-equal" {
        description "Indicates that the comparison type is greater than or equal to.";
    }
}

description "Comparison type.";

leaf value {
    type match-value;
    description "Comparison value which is static threshold value.";
}

leaf target {
    type target;
    description "List for target management objects.";
}

description "Container for boolean test.";

container threshold {
    leaf rising-value {
        type match-value;
        description
        "Indicates that the comparison type is unequal to.";
    }
}
"Sets the rising threshold to the specified value, when the current sampled value is greater than or equal to this threshold, and the value at the last sampling interval was less than this threshold, the event is triggered."

leaf-list rising-target {
  type target;
  description
    "List for target objects.";
}
leaf falling-value {
  type match-value;
  description
    "Sets the falling threshold to the specified value.";
}
leaf-list falling-target {
  type target;
  description
    "List for target objects.";
}
leaf delta-rising-value {
  type match-value;
  description
    "Sets the delta rising threshold to the specified value.";
}
leaf-list delta-rising-target {
  type target;
  description
    "List for target objects.";
}
leaf delta-falling-value {
  type match-value;
  description
    "Sets the delta falling threshold to the specified value.";
}
leaf-list delta-falling-target {
  type target;
  description
    "List for target objects.";
}
leaf startup {
  type enumeration {
    enum "rising" {
      description
        "If the first sample after this managed object becomes active is greater than or equal to ‘rising-value’ and the ‘startup’ is equal to ‘rising’ then one threshold rising event is
triggered for that managed object.

} enum "falling" {
  description
  "If the first sample after this managed object becomes
  active is less than or equal to 'falling-value' and
  the 'startup' is equal to 'falling' then one
  threshold falling event is triggered for that managed
  object.";
}

} enum "rising-or-falling" {
  description
  "That event may be triggered when the
  'startup' is equal to 'rising-or-falling'.
  'rising-or-falling' indicate the state value of the
  managed object may less than or greater than the
  specified threshold value.";
}

} description
  "Startup setting.";
}

} description
  "Container for the threshold trigger condition.
  Note that the threshold here may change over time
  or the state value changes in either ascen order
  or descend order.";
}

} module ietf-event {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-event";
  prefix evt;

  import ietf-yang-types {
    prefix yang;
  }

  import ietf-event-trigger {

  

<CODE ENDS>
prefix trig;
}

organization
  "IETF xxx Working Group";
contact
  "Zitao Wang: wangzitao@huawei.com
   Qin Wu: bill.wu@huawei.com";
description
  "This module defines a model for the service topology."

revision 2018-12-18 {
  description
    "Initial revision."
  reference "foo";
}

identity event-type {
  description
    "Base identity for event type"
}

identity frequency {
  description
    "Base identity for frequency"
}

identity periodic {
  base frequency;
  description
    "Identity for periodic trigger"
}

identity scheduling {
  base frequency;
  description
    "Identity for scheduling trigger"
}

typedef match-value {
  type union {
    type yang:xpath1.0;
    type yang:object-identifier;
    type string;
  }
  description
    "This type is used to match resources of type ‘target’. Since the type ‘target’ is a union of different types,
typedef target {
  type union {
    type instance-identifier;
    type yang:object-identifier;
    type yang:uuid;
    type string;
  }
  description
  "If the target is modelled in YANG, this type will
  be an instance-identifier.
  If the target is an SNMP object, the type will be an
  object-identifier.
  If the target is anything else, for example a distinguished
  name or a CIM path, this type will be a string.
  If the target is identified by a UUID use the uuid
  type.
  If the server supports several models, the precedence should
  be in the order as given in the union definition.";
}

grouping start-end-grouping {
  description
  "A grouping that provides start and end times for
  Event objects."
  leaf start {
    type yang:date-and-time;
    description
    "The date and time when the Event object
    starts to create triggers.";
  }
  leaf end {
    type yang:date-and-time;
    description
    "The date and time when the Event object
    stops to create triggers.
    It is generally a good idea to always configure
    an end time and to refresh the end time as needed
    to ensure that agents that lose connectivity to
    their Controller do not continue executing Schedules
    forever.";
  }
}

container events {

}
leaf min-data-object {
    type uint32;
    description
        "Sets the minimum number for a set of data collected or selected during the service monitoring."
}
leaf max-data-object {
    type uint32;
    description
        "Sets the maximum number for a set of data collected or selected during the service monitoring.";
}
list event {
    key "event-name type";
    leaf event-name {
        type string;
        description
            "Event name";
    }
    leaf type {
        type identityref {
            base event-type;
        }
        description
            "Type of event";
    }
    leaf event-description {
        type string;
        description
            "Event description";
    }
    leaf-list target {
        type target;
        description
            "targeted objects";
    }
    leaf clear {
        type boolean;
        default "false";
        description
            "A flag indicate whether the event be closed";
    }
    list related-event {
        key "event-name type";
        leaf event-name {
            type string;
            description
                "Event name";
    }
leaf type {
    type identityref {
        base event-type;
    }
    description
        "Type of event";
}

description
    "List for related events";
}

uses trig:trigger-grouping;

list action {
    key "action-name";
    leaf action-name {
        type string;
        description
            "Action name";
    }
}

notification event-notification {
    leaf event-name {
        type leafref {
            path "/events/event/event-name";
        }
        description
            "Report the event name";
    }
    leaf type {
        type leafref {
            path "/events/event/type";
        }
        description
            "Report the event type";
    }
    leaf-list target {
        type target;
        description
            "Report the target objects";
    }
    description
        "This notification is used to report that an operator
         acted upon an Event.";
}

action set {
    input {
        leaf-list target {
9. Security Considerations

The YANG modules defined in this document MAY be accessed via the
RESTCONF protocol [RFC8040] or NETCONF protocol ([RFC6241]). The
lowest RESTCONF or NETCONF layer requires that the transport-layer
protocol provides both data integrity and confidentiality, see
Section 2 in [RFC8040] and [RFC6241]. The lowest NETCONF layer is
the secure transport layer, and the mandatory-to-implement secure
transport is Secure Shell (SSH) [RFC6242]. The lowest RESTCONF layer
is HTTPS, and the mandatory-to-implement secure transport is TLS
[RFC5246].

The NETCONF access control model [RFC6536] 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.
There are a number of data nodes defined in this YANG module that are writable/creatable/deletable (i.e., config true, which is the default). These data nodes may be considered sensitive or vulnerable in some network environments. Write operations (e.g., edit-config) to these data nodes without proper protection can have a negative effect on network operations. These are the subtrees and data nodes and their sensitivity/vulnerability:

- /events/event/event-name
- /events/event/target
- /events/action/set/target
- /events/event/trigger/name

10. IANA Considerations

This document registers a URI in the IETF XML registry [RFC3688]. Following the format in [RFC3688], the following registration is requested to be made:

```
---

Registrant Contact: The IESG.
XML: N/A, the requested URI is an XML namespace.
---
```

This document registers a YANG module in the YANG Module Names registry [RFC6020].

```
---

Name:         ietf-event
Prefix:       evt
Reference:    RFC xxxx
---
```

11. Normative References


Appendix A. Example of Event

For example, some service requires monitoring the "in-errors" state of the interface, and if the value of "in-errors" exceeds the threshold, the event should reset the interface’s enabled value to false:

<events>
  <event>
    <event-name>interface-state-exception</event-name>
    <type>interface-exception</type>
    <target>/if:interfaces/if:interface[if:name='eth1']</target>
    <target>/if:interfaces/if:interface[if:name='eth2']</target>
    <target>/if:interfaces/if:interface[if:name='eth3']</target>
    <trigger>
      <name>evaluate-in-errors</name>
      <trigger-description>evaluate the number of packets that contained errors</trigger-description>
      <frequency>10m</frequency>
      <type>threshold</type>
      <test>
        <threshold>
          <startup>rising</startup>
          <rising-value>100</rising-value>
          <rising-target>/if:interfaces/if:interface[if:name='eth1']
          /if:statistic/if:in-errors</rising-target>
          <rising-target>/if:interfaces/if:interface[if:name='eth2']
          /if:statistic/if:in-errors</rising-target>
          <rising-target>/if:interfaces/if:interface[if:name='eth3']
          /if:statistic/if:in-errors</rising-target>
        </threshold>
      </test>
    </trigger>
    <action>
      <name>interface-exception</name>
      <event-notification>
        <event-name>interface-state-exception</event-name>
        <type>interface-exception</type>
        <target>/if:interfaces/if:interface[if:name='eth1']</target>
      </event-notification>
      <set>
        <target>/if:interfaces/if:interface[if:name='eth1']</target>
        <integer-value>
          <interfaces>
            <interface>
              <name>eth1</name>
              <enable>false</enable>
            </interface>
          </interfaces>
        </integer-value>
      </set>
    </action>
  </event>
</events>
Appendix B. Changes between revisions

v00 - v01

- Separate ietf-event-trigger.yang from Event management model and ietf-event.yang and make it reusable in other YANG models.
- Clarify the difference between boolean trigger condition and threshold trigger condition.
- Change evt-smp-min and evt-smp-max into min-data-object and max-data-object in the data model.

Authors’ Addresses

Michael Wang
Huawei Technologies, Co., Ltd
101 Software Avenue, Yuhua District
Nanjing 210012
China

Email: wangzitao@huawei.com

Qin Wu
Huawei
101 Software Avenue, Yuhua District
Nanjing, Jiangsu 210012
China

Email: bill.wu@huawei.com

Chongfeng Xie
China Telecom

Email: xiechf@ctbri.com.cn