Abstract

This document describes a data model for configuration scheduling.

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), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

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."

The list of current Internet-Drafts can be accessed at
http://www.ietf.org/ietf/1id-abstracts.txt

The list of Internet-Draft Shadow Directories can be accessed at
http://www.ietf.org/shadow.html

This Internet-Draft will expire on April 24, 2017.
1. Introduction

This document introduces a YANG [RFC6020] data model for configuration scheduling. This model can be used together with other YANG data models to specify a schedule applied on a configuration data node, so that the configuration data can take effect according to the schedule.

1.1. Terminology

The keywords "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].

The following terms are defined in [RFC6020] and are not redefined here:
2. Motivation

Some applications benefit from resource scheduling to allow operators to plan ahead of time. Traffic engineering is one of such examples [RFC7399]. When configuration and state models are designed for such applications, it has been considered that certain data objects need to be configured according to predefined schedules. In other situations, operators need to de-configure certain data objects at predefined schedules for the purposes such as maintenance. These data objects are interpreted and implemented by the applicable applications.

3. Configuration Scheduling YANG Data Model Overview

This document defines a YANG data model that specifies configuration schedules for other YANG data models. For each targeted configuration data object or a group of configuration data objects, an entry is specified along with requested schedules using this configuration schedule model. The application implementing the targeted schema nodes implements the configuration schedules, configuring or de-configuring the specified objects according to the specified schedules. The model schema of the targeted application does not need changes, so the data model described in this document can be used for any data model. The configuration scheduling YANG data model has the following structure:

```yang
module ietf-schedule {
  +--rw configuration-schedules
      +--rw target* [object]
          +--rw object yang:xpath1.0
          +--rw data-value? string
          +--rw schedules
              +--rw schedule* [schedule-id] {
                  +--rw schedule-id uint32
                  +--rw inclusive-exclusive? enumeration
                  +--rw start? yang:date-and-time
                  +--rw schedule-duration? string
                  +--rw repeat-interval? string
              }
        }
    }
```
++ro next-event
++ro start?       yang:date-and-time
++ro duration?    string
++ro operation?   enumerationmodule: ietf-schedule

4. Usage Example

The following model defines a list of TE (Traffic Engineering) links which can be configured with specified schedules:

module: example
++rw te-links
   ++rw te-link* [id]
      ++rw id           string

The following configuration requests that

  o link-1 is configured weekly for five one-day periods, starting from 2016-09-12T23:20:50.52Z.

  o link-2 is de-configured for two hours, starting from 2016-09-15T01:00:00.00Z.

<configuration-schedules>
  <target xmlns:ex="urn:example">
    <object>/ex:te-links/ex:te-link[ex:link-id='link-1']</object>
    <schedules>
      <schedule>
        <schedule-id>11</schedule-id>
        <start>2016-09-12T23:20:50.52Z</start>
        <schedule-duration>P1D</schedule-duration>
        <repeat-interval>R5/P1W</repeat-interval>
      </schedule>
    </schedules>
  </target>
  <target xmlns:ex="urn:example">
    <object>/ex:te-links/ex:te-link[ex:link-id='link-2']</object>
    <schedules>
      <schedule>
        <schedule-id>12</schedule-id>
        <inclusive-exclusive>exclusive</inclusive-exclusive>
        <start>2016-09-15T01:00:00.00Z</start>
        <schedule-duration>P2H</schedule-duration>
      </schedule>
    </schedules>
  </target>
</configuration-schedules>
5. Configuration Scheduling YANG Module

```yang
<CODE BEGINS> file "ietf-schedule@2016-10-11.yang"
module ietf-schedule {
  yang-version 1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-schedule";
  // replace with IANA namespace when assigned
  prefix "sch";

  import ietf-yang-types {
    prefix "yang";
  }

  organization "TBD";
  contact "TBD";
  description
    "The model allows time scheduling parameters to be specified.";

  revision "2016-10-11" {
    description "Initial revision";
    reference "TBD";
  }

  /*
   * Groupings
   */

  grouping schedule-config-attributes {
    description
      "A group of attributes for a schedule.";

    leaf inclusive-exclusive {
      type enumeration {
        enum inclusive {
          description "The schedule element is inclusive.";
        }
      }
    }

Liu, et al.  Expires April 24, 2017
```
enum exclusive {
  description "The schedule element is exclusive.";
}

default "inclusive";
description "Whether the list item is inclusive or exclusive.";

leaf start {
  type yang:date-and-time;
description "Start time.";
}

leaf schedule-duration {
  type string {
    pattern
    'P(\d+Y)?(\d+M)?(\d+W)?(\d+D)?T(\d+H)?(\d+M)?(\d+S)?';
  }
description "Schedule duration in ISO 8601 format.";
}

leaf repeat-interval {
  type string {
    pattern
    'R\d*/P(\d+Y)?(\d+M)?(\d+W)?(\d+D)?T(\d+H)?(\d+M)?'
    + '(%d+S)?';
  }
description "Repeat interval in ISO 8601 format.";
}

// schedule-config-attributes

grouping schedule-state-attributes {
  description "State attributes for a schedule.";
  container next-event {
    description "The state information of the next scheduled event.";
    leaf start {
      type yang:date-and-time;
description "Start time.";
    }
    leaf duration {

type string {
    pattern
        'P(\d+Y)?(\d+M)?(\d+W)?(\d+D)?T(\d+H)?(\d+M)?(\d+S)\''
    description "Schedule duration in ISO 8601 format.";
}
leaf operation {
    type enumeration {
        enum configure {
            description
                "Create the configuration data.";
        }
        enum deconfigure {
            description
                "Remove the configuration data.";
        }
        enum set {
            description
                "Set the specified configuration data.";
        }
        enum reset {
            description
                "Revert the specified configuration data back to the
                original value.";
        }
    }
    description "Operation type.";
}
} // next-event
} // schedule-state-attributes

grouping schedules {
    description
        "A list of schedules defining when a particular
        configuration takes effect.";
    container schedules {
        description
            "Container of a schedule list defining when a particular
            configuration takes effect.";
    list schedule {
        key "schedule-id";
    }
}
description "A list of schedule elements.";
leaf schedule-id {
    type uint32;
    description "Identifies the schedule element.";
}
uses schedule-config-attributes;
}
} // schedules

/*
 * Configuration data nodes
 */
container configuration-schedules {
    description "Serves as top-level container for a list of configuration schedules.";
    list target {
        key "object";
        description "A list of targets that configuration schedules are applied.";
        leaf object {
            type yang:xpath1.0;
            description "Xpath defining the data items of interest.";
        }
        leaf data-value {
            type string;
            description "The ephemeral value applied to the leaf data node specified by data-objects. This is applicable when object is a leaf.";
        }
        uses schedules;
        container state {
            config false;
            description "Operational state data.";
            uses schedule-state-attributes;
        } // state
6. Security Considerations

The configuration, state, action and notification data defined in this document are designed to be accessed via the NETCONF protocol [RFC6241]. The data-model by itself does not create any security implications. The security considerations for the NETCONF protocol are applicable. The NETCONF protocol used for sending the data supports authentication and encryption.

7. References

7.1. Normative References


7.2. Informative References

Authors’ Addresses

Xufeng Liu
Kuatro Technologies
8281 Greensboro Drive, Suite 200
McLean, VA 22102
USA

Email: xliu@kuatrotech.com

Igor Bryskin
Huawei Technologies
Email: Igor.Bryskin@huawei.com

Vishnu Pavan Beeram
Juniper Networks
Email: vbeeram@juniper.net

Tarek Saad
Cisco Systems Inc
Email: tsaad@cisco.com

Himanshu Shah
Ciena
Email: hshah@ciena.com

Oscar Gonzalez de Dios
Telefonica
Email: oscar.gonzalezdedios@telefonica.com