

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: August 19, 2013

J. Schoenwaelder
Jacobs University Bremen
February 15, 2013

A YANG Data Model for LMAP Measurement Agents
draft-schoenw-lmap-yang-00.txt

Abstract

This document sketches a data model for configuring and scheduling tests for large scale broadband access network measurements. The data model is defined using the YANG data modeling language.

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 <http://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 August 19, 2013.

Copyright Notice

Copyright (c) 2013 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 (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1.	Introduction	3
2.	Data Model Overview	3
3.	YANG Module	4
4.	Security Considerations	9
5.	IANA Considerations	9
6.	Acknowledgements	9
7.	References	9
7.1.	Normative References	9
7.2.	Informative References	10
	Author's Address	10

1. Introduction

This document sketches a data model for configuring and scheduling tests for large scale broadband access network measurements. The data model is defined using the YANG [\[RFC6020\]](#) data modeling language.

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 [\[RFC2119\]](#).

2. Data Model Overview

The data model has the following structure, where square brackets are used to enclose a list's keys, "?" means that the leaf is optional, and "*" denotes a leaf-list:

```
module: acme-lmap
  +--rw lmap
    +--rw tests
      | +--rw test [name]
      |   +--rw name          string
      |   +--rw description?  string
      |   +--rw program       string
      |   +--rw option [name]
      |     | +--rw name      string
      |     | +--rw value?   string
      |     +--rw argument*  string
    +--rw schedules
      +--rw schedule [name]
        +--rw name          string
        +--rw test?         leafref
        +--rw enabled?      boolean
        +--rw (schedule-type)?
          | +--:(periodic)
          | | +--rw interval?    uint32
          | +--:(calendar)
          | | +--rw weekday?     weekday-set
          | | +--rw month-set?   months-set
          | | +--rw day*         int8
          | | +--rw hour*        int8
          | | +--rw minute*      int8
          | +--:(one-shot)
        +--ro failures?      yang:counter32
        +--ro last-failure?  string
        +--ro last-failed?   yang:date-and-time
```


3. YANG Module

The data model uses types imported from [[RFC6021](#)].

```
module acme-lmap {  
  
  namespace "http://www.example.org/yang/acme-lmap";  
  
  prefix "lmap";  
  
  import ietf-yang-types {  
    prefix yang;  
  }  
  
  organization  
    "Large-Scale Measurement of Broadband Performance Project";  
  
  contact  
    "Web:      <http://www.example.org/>  
  
    Editor:   Juergen Schoenwaelder  
              <j.schoenwaelder@jacobs-university.de>";  
  
  description  
    "This module provides a data model for configuring tests and  
    test schedules running on LMAP measurement agents.  
  
    Note that the whole data model can be generalized to invoke  
    other actions that are not measurement tests.";  
  
  revision "2013-02-01" {  
    description  
      "Initial version";  
    reference  
      "TBD";  
  }  
  
  typedef weekday-set {  
    type bits {  
      bit sunday;  
      bit monday;  
      bit tuesday;  
      bit wednesday;  
      bit thursday;  
      bit friday;  
      bit saturday;  
    }  
    description
```



```
"A type modeling sets of weekdays in the Greco-Roman
  tradition.";
}

typedef months-set {
  type bits {
    bit january;
    bit february;
    bit march;
    bit april;
    bit may;
    bit june;
    bit july;
    bit august;
    bit september;
    bit october;
    bit november;
    bit december;
  }
  description
    "A type modeling sets of months in the Julian and Gregorian
    tradition.";
}

container lmap {

  container tests {
    config true;

    list test {
      key name;

      description
        "The list of tests configured on the lmap probe.";

      leaf name {
        type string;
        description
          "The unique name of a configured test.";
      }

      leaf description {
        type string;
        description
          "A short description of the configured test.";
      }

      leaf program {
```



```
    type string;
    mandatory true;
    description
      "The (local) program to invoke in order to execute
        the test.";
  }

  list option {
    key "name";
    ordered-by user;

    description
      "The options passed to the program in order to carry out
        the test. This is a list of key / value pairs and may be
        used to model command line options.";

    leaf name {
      type string;
      description
        "The name of the options.";
    }

    leaf value {
      type string;
      description
        "The value of the options.";
    }
  }

  leaf-list argument {
    type string;
    description
      "The list of arguments passed to the test.";
  }
}

container schedules {
  config true;

  list schedule {
    key name;

    leaf name {
      type string;
      description
        "The locally-unique, administratively assigned name for
          this scheduled test.";
    }
  }
}
```



```
}

leaf test {
  type leafref {
    path "/lmap/tests/test/name";
  }
  description
    "The test invoked by this schedule";
}

leaf enabled {
  type boolean;
  description
    "Indicates whether the test is enabled or disabled.";
}

choice schedule-type {

  case periodic {
    leaf interval {
      type uint32;
      units "seconds";
      description
        "The number of seconds between two action invocations of
        a periodic test. Implementations must guarantee
        that test invocations will not occur before at least
        the specified number of seconds have passed.

        The scheduler must ignore all periodic schedules that
        have a interval value of 0. A periodic schedule
        with a scheduling interval of 0 seconds will therefore
        never invoke a test.

        Implementations may be forced to delay invocations in
        the face of local constraints. A scheduled test should
        therefore not rely on the accuracy provided by the
        scheduler implementation.";
    }
  }

  case calendar {
    leaf weekday {
      type weekday-set;
      description
        "The set of weekdays on which this test should be
        executed. And empty set means any weekday.";
    }
  }
}
```



```
leaf month-set {
  type months-set;
  description
    "The set of months on which this test should be
    executed. And empty set means any month.";
}

leaf-list day {          // make this a set?
  type int8 {
    range "-31..31";
  }
  description
    "The set of days in a months on which this test should
    be executed. Negative days are counted backwards from
    the end of the month.";
}

leaf-list hour {         // make this a set?
  type int8 {
    range "0..23";
  }
  description
    "The set of hours in a day on which this test should
    be executed.";
}

leaf-list minute {       // make this a set?
  type int8 {
    range "0..59";
  }
  description
    "The set of minutes in an hour on which this test
    should be executed.";
}

case one-shot {          // separate or add disable counter
}                          // to the calendar schedule?
}

leaf failures {
  config false;
  type yang:counter32;
  description
    "The number of failures that occurred while invoking
    scheduled tests.";
}
```



```
    leaf last-failure {
      config false;
      type string;
      description
        "The most recent error that occurred during the invocation of
        a scheduled test.";
    }

    leaf last-failed {
      config false;
      type yang:date-and-time;
      description
        "The date and time when the most recent failure occurred.";
    }
  }
}
}
```

4. Security Considerations

TBD

5. IANA Considerations

TBD

6. Acknowledgements

Some ideas were lifted from [[RFC2591](#)].

7. References

7.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), March 1997.
- [RFC6020] Bjorklund, M., "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", [RFC 6020](#), October 2010.

[RFC6021] Schoenwaelder, J., "Common YANG Data Types", [RFC 6021](#), October 2010.

[7.2.](#) Informative References

[RFC2591] Levi, D. and J. Schoenwaelder, "Definitions of Managed Objects for Scheduling Management Operations", [RFC 2591](#), May 1999.

Author's Address

Juergen Schoenwaelder
Jacobs University Bremen

Email: j.schoenwaelder@jacobs-university.de

