Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: June 17, 2016

A. Bierman
YumaWorks
M. Bjorklund
Tail-f Systems
K. Watsen
Juniper Networks
December 15, 2015

YANG Module Library
draft-ietf-netconf-yang-library-03

Abstract

This document describes a YANG library, which provides information about all the YANG modules used by a device to represent management and protocol information. A YANG library can be shared by multiple protocols within the same device. Simple caching mechanisms are needed to allow clients to minimize retrieval of this information.

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 June 17, 2016.

Copyright Notice

Copyright (c) 2015 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 2
1.1. Terminology 3
1.1.1. NETCONF 3
1.1.2. YANG 3
1.1.3. Terms 3
1.1.4. Tree Diagrams 4
2. YANG Module Library 4
2.1. modules 5
2.1.1. modules/module-set-id 5
2.1.2. modules/yang-protocol 5
2.1.3. modules/module 5
2.2. YANG Library Module 6
3. IANA Considerations 12
3.1. YANG Module Registry 12
4. Security Considerations 12
5. Acknowledgements 13
6. Normative References 13
Appendix A. Change Log 13
A.1. v02 to v03 13
A.2. v01 to v02 14
A.3. v00 to v01 14
A.4. draft-ietf-netconf-restconf-03 to v00 14
Appendix B. Open Issues 14
Authors' Addresses 14

1. Introduction

There is a need for standard mechanisms to identify the YANG modules and submodules that are in use by a server that utilizes YANG-based data abstractions. If a large number of YANG modules are utilized by the server, then the YANG library information needed can be relatively large. This information changes very infrequently, so it is important that clients be able to cache the YANG library and easily identify if their cache is out-of-date.

YANG library information can be different on every server, and can change at run-time or across a server reboot. Typically, a firmware upgrade is required to change the set of YANG modules used by a server.

The following information is needed by a client application (for each YANG module in the library) to fully utilize the YANG data modeling language:

- o name: The mandatory YANG module name MUST be unique within a YANG library.

- o revision: Each YANG module and submodule within the library has a revision. This is derived from the most recent revision statement within the module or submodule. If no such revision statement exists, the module's or submodule's revision is the empty string.

- o submodule list: The name and revision of each submodule used by the module MUST be identified.

- o feature list: The name of each YANG feature supported by the server MUST be identified.

- o deviation list: The name of each YANG module used for deviation statements SHOULD be identified.

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].

1.1.1. NETCONF

The following terms are defined in [RFC6241]:

- o client

- o server

1.1.2. YANG

The following terms are defined in [RFC6020]:

- o module

- o submodule

1.1.3. Terms

The following terms are used within this document:

- o YANG library: a collection of YANG modules and submodules used by a server

1.1.4. Tree Diagrams

A simplified graphical representation of the data model is used in this document. The meaning of the symbols in these diagrams is as follows:

- o Brackets "[" and "]" enclose list keys.
- o Abbreviations before data node names: "rw" means configuration data (read-write) and "ro" state data (read-only).
- o Symbols after data node names: "?" means an optional node, "!" means a presence container, and "*" denotes a list and leaf-list.
- o Parentheses enclose choice and case nodes, and case nodes are also marked with a colon (":").
- o Ellipsis ("...") stands for contents of subtrees that are not shown.

2. YANG Module Library

The "ietf-yang-library" module provides information about the YANG library used by a server.

YANG Tree Diagram for "ietf-yang-library" module:

```
+--ro modules
   +--ro module-set-id    string
   +--ro yang-protocol*   identityref
   +--ro module* [name revision]
      +--ro name                   yang:yang-identifier
      +--ro revision               union
      +--ro schema?                inet:uri
      +--ro namespace              inet:uri
      +--ro feature*               yang:yang-identifier
      +--ro deviation* [name revision]
      |  +--ro name        yang:yang-identifier
      |  +--ro revision    union
      +--ro conformance            enumeration
      +--ro submodules
      |  +--ro submodule* [name revision]
      |     +--ro name        yang:yang-identifier
      |     +--ro revision    union
      |     +--ro schema?     inet:uri
      +--ro restricted-protocol*   identityref
```

2.1. modules

This mandatory container holds the identifiers for the YANG data model modules supported by the server.

2.1.1. modules/module-set-id

This mandatory leaf contains a unique implementation-specific identifier representing the current set of modules and submodules. This can for example be a checksum of all modules and submodules.

This leaf allows a client to fetch the module list once, cache them, and only re-fetch them if the value of this leaf has been changed.

2.1.2. modules/yang-protocol

This leaf-list identifies the YANG-based protocols that are using the YANG library. It allows an operator to be aware that multi-protocol conflicts could potentially occur on a managed device.

A corresponding leaf-list called 'restricted-protocol' is defined in the module entry to identify modules that are not supported by one or more protocols.

2.1.3. modules/module

This mandatory list contains one entry for each YANG data model module supported by the server. There MUST be an entry in this list for every YANG module that is used by the server.

2.2. YANG Library Module

The "ietf-yang-library" module defines monitoring information for the YANG modules used by a server.

The "ietf-yang-types" and "ietf-inet-types" modules from [RFC6991] are used by this module for some type definitions.

RFC Ed.: update the date below with the date of RFC publication and remove this note.

```
<CODE BEGINS> file "ietf-yang-library@2015-10-18.yang"

module ietf-yang-library {
  namespace "urn:ietf:params:xml:ns:yang:ietf-yang-library";
  prefix "yanglib";

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

  organization
    "IETF NETCONF (Network Configuration) Working Group";

  contact
    "WG Web:   <http://tools.ietf.org/wg/netconf/>
     WG List:  <mailto:netconf@ietf.org>

     WG Chair: Mehmet Ersue
               <mailto:mehmet.ersue@nsn.com>

     WG Chair: Mahesh Jethanandani
               <mailto:mjethanandani@gmail.com>

     Editor:   Andy Bierman
               <mailto:andy@yumaworks.com>

     Editor:   Martin Bjorklund
               <mailto:mbj@tail-f.com>

     Editor:   Kent Watsen
```

```
                 <mailto:kwatsen@juniper.net>";

     description
       "This module contains monitoring information about the YANG
        modules and submodules that are used within a YANG-based
        server.

        Copyright (c) 2015 IETF Trust and the persons identified as
        authors of the code.  All rights reserved.

        Redistribution and use in source and binary forms, with or
        without modification, is permitted pursuant to, and subject
        to the license terms contained in, the Simplified BSD License
        set forth in Section 4.c of the IETF Trust's Legal Provisions
        Relating to IETF Documents
        (http://trustee.ietf.org/license-info).

        This version of this YANG module is part of RFC XXXX; see
        the RFC itself for full legal notices.";

     // RFC Ed.: replace XXXX with actual RFC number and remove this
     // note.

     // RFC Ed.: remove this note
     // Note: extracted from draft-ietf-netconf-yang-library-03.txt

     // RFC Ed.: update the date below with the date of RFC publication
     // and remove this note.
     revision 2015-12-08 {
       description
         "Initial revision.";
       reference
         "RFC XXXX: YANG Module Library.";
     }

     typedef revision-identifier {
       type string {
         pattern '\d{4}-\d{2}-\d{2}';
       }
       description
         "Represents a specific date in YYYY-MM-DD format.";
     }

     identity yang-protocol {
       description
        "The base for all YANG-based protocol identifiers.";
     }
```

```
     identity netconf1.0 {
       base yang-protocol;
       description "Version 1.0 of the NETCONF protocol";
       reference "RFC 4271";
     }

     identity netconf1.1 {
       base yang-protocol;
       description "Version 1.1 of the NETCONF protocol";
       reference "RFC 6241";
     }

     identity restconf1.0 {
       base yang-protocol;
       description "Version 1.0 of the RESTCONF protocol";
       reference "draft-ietf-netconf-restconf";
     }

     grouping module-list {
       description
         "The module data structure is represented as a grouping
          so it can be reused in configuration or another monitoring
          data structure.";

       grouping common-leafs {
         description
           "Common parameters for YANG modules and submodules.";

         leaf name {
           type yang:yang-identifier;
           description
             "The YANG module or submodule name.";
         }
         leaf revision {
           type union {
             type revision-identifier;
             type string { length 0; }
           }
           description
             "The YANG module or submodule revision date.
              An empty string is used if no revision statement
              is present in the YANG module or submodule.";
         }
       }

       grouping schema-leaf {
         description
           "Common schema leaf parameter for modules and submodules.";
```

```
          leaf schema {
            type inet:uri;
            description
              "Contains a URL that represents the YANG schema
               resource for this module or submodule.

               This leaf will only be present if there is a URL
               available for retrieval of the schema for this entry.";
          }
        }

        list module {
          key "name revision";
          description
            "Each entry represents one module currently
             supported by the server.";

          uses common-leafs;
          uses schema-leaf;

          leaf namespace {
            type inet:uri;
            mandatory true;
            description
              "The XML namespace identifier for this module.";
          }
          leaf-list feature {
            type yang:yang-identifier;
            description
              "List of YANG feature names from this module that are
               supported by the server.";
          }
          list deviation {
            key "name revision";
            description
              "List of YANG deviation module names and revisions
               used by this server to modify the conformance of
               the module associated with this entry.  Note that
               the same module can be used for deviations for
               multiple modules, so the same entry MAY appear
               within multiple 'module' entries.

               If the deviation module is available for download
               from the server then a 'module' entry for that module
               will exist, with the same name and revision values.
               The 'conformance' value will be 'implement' for
               the deviation module.";
            uses common-leafs;
```

```
        }
        leaf conformance {
          type enumeration {
            enum implement {
              description
                "Indicates that the server implements one or more
                 protocol-accessible objects defined in the YANG module
                 identified in this entry.  This includes deviation
                 statements defined in the module.

                 For YANG 1.1 modules, there MUST NOT be more than one
                 module entry with conformance 'implement' for a
                 particular module name.

                 For YANG 1.1 modules that use the import statement
                 without specifying a revision date, the implemented
                 revision of the imported module MUST be used.
                 If the imported module is not implemented, then
                 the most recent revision of the imported module
                 used by the server (and contained in the module list)
                 MUST be used.

                 For YANG 1.0 modules, there SHOULD NOT be more than
                 one module entry for a particular module name.";
            }
            enum import {
              description
                "Indicates that the server imports reusable definitions
                 from the specified revision of the module, but does
                 not implement any protocol accessible objects from
                 this revision.

                 Multiple module entries for the same module name MAY
                 exist. This can occur if multiple modules import the
                 same module, but specify different revision-dates in
                 the import statements.

                 For import statements that do not specify a revision
                 date, the most recent revision in the library SHOULD
                 be used by the server.";
            }
          }
          mandatory true;
          description
            "Indicates the type of conformance the server is claiming
             for the YANG module identified by this entry.";
        }
        container submodules {
```

```
          description
            "Contains information about all the submodules used
             by the parent module entry";

          list submodule {
            key "name revision";
            description
              "Each entry represents one submodule within the
               parent module.";
            uses common-leafs;
            uses schema-leaf;
          }
        }
        leaf-list restricted-protocol {
          type identityref {
            base yang-protocol;
          }
          description
           "Identifies a protocol that does not use the module
            described by this module instance.";
        }
      } // list module
    }  // grouping module


    container modules {
      config false;
      description
        "Contains YANG module monitoring information.";

      leaf module-set-id {
        type string;
        mandatory true;
        description
          "Contains a server-specific identifier representing
           the current set of modules and submodules.  The
           server MUST change the value of this leaf if the
           information represented by the 'module' list instances
           has changed.";
      }

      leaf-list yang-protocol {
        type identityref {
          base yang-protocol;
        }
        description
         "Identifies a protocol that is using modules described
          in this library.  There SHOULD be one instance of
```

```
          this object for each protocol for each YANG-based
          protocol using this library.";
      }

      uses module-list;
    }

}

<CODE ENDS>
```

3. IANA Considerations

3.1. YANG Module Registry

This document registers one URI in the IETF XML registry [RFC3688]. Following the format in RFC 3688, the following registration is requested to be made.

```
URI: urn:ietf:params:xml:ns:yang:ietf-yang-library
Registrant Contact: The NETMOD WG of the IETF.
XML: N/A, the requested URI is an XML namespace.
```

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

```
name:         ietf-yang-library
namespace:    urn:ietf:params:xml:ns:yang:ietf-yang-library
prefix:       yanglib
// RFC Ed.: replace XXXX with RFC number and remove this note
reference:    RFC XXXX
```

4. Security Considerations

The YANG module defined in this memo is designed to be accessed via the NETCONF protocol [RFC6241]. The lowest NETCONF layer is the secure transport layer and the mandatory-to-implement secure transport is SSH [RFC6242].

Some of the readable data nodes in this YANG module may be considered sensitive or vulnerable in some network environments. It is thus important to control read access (e.g., via get, get-config, or notification) to these data nodes. These are the subtrees and data nodes and their sensitivity/vulnerability:

o /modules/module: The module list used in a server implementation may help an attacker identify the server capabilities and server implementations with known bugs. Server vulnerabilities may be

specific to particular modules, module revisions, module features, or even module deviations. This information is included in each module entry. For example, if a particular operation on a particular data node is known to cause a server to crash or significantly degrade device performance, then the module list information will help an attacker identify server implementations with such a defect, in order to launch a denial of service attack on the device.

5. Acknowledgements

Contributions to this material by Andy Bierman are based upon work supported by the The Space & Terrestrial Communications Directorate (S&TCD) under Contract No. W15P7T-13-C-A616. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of The Space & Terrestrial Communications Directorate (S&TCD).

6. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC3688] Mealling, M., "The IETF XML Registry", BCP 81, RFC 3688, January 2004.

[RFC6020] Bjorklund, M., "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", RFC 6020, October 2010.

[RFC6241] Enns, R., Ed., Bjorklund, M., Ed., Schoenwaelder, J., Ed., and A. Bierman, Ed., "Network Configuration Protocol (NETCONF)", RFC 6241, June 2011.

[RFC6242] Wasserman, M., "Using the NETCONF Protocol over Secure Shell (SSH)", RFC 6242, June 2011.

[RFC6991] Schoenwaelder, J., "Common YANG Data Types", RFC 6991, July 2013.

Appendix A. Change Log

-- RFC Ed.: remove this section before publication.

A.1. v02 to v03

o added yang-protocol identity

- o added identities for NETCONF and RESTCONF protocols
- o added yang-protocol leaf-list to /modules
- o added restricted-protocol leaf-list to /modules/module

A.2. v01 to v02

- o clarify 'implement' conformance for YANG 1.1 modules

A.3. v00 to v01

- o change conformance leaf to enumeration
- o filled in security considerations section

A.4. draft-ietf-netconf-restconf-03 to v00

- o moved ietf-yang-library from RESTCONF draft to new draft

Appendix B. Open Issues

-- RFC Ed.: remove this section before publication.

The YANG Library issue tracker can be found here:

https://github.com/netconf-wg/yang-library/issues [1]

Authors' Addresses

Andy Bierman
YumaWorks

Email: andy@yumaworks.com

Martin Bjorklund
Tail-f Systems

Email: mbj@tail-f.com

Kent Watsen
Juniper Networks

Email: kwatsen@juniper.net