

Internet Engineering Task Force (IETF)
Request for Comments: 7895
Category: Standards Track
ISSN: 2070-1721

A. Bierman
YumaWorks
M. Bjorklund
Tail-f Systems
K. Watsen
Juniper Networks
June 2016

YANG Module Library

Abstract

This document describes a YANG library that provides information about all the YANG modules used by a network management server (e.g., a Network Configuration Protocol (NETCONF) server). Simple caching mechanisms are provided to allow clients to minimize retrieval of this information.

Status of This Memo

This is an Internet Standards Track document.

This document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and has been approved for publication by the Internet Engineering Steering Group (IESG). Further information on Internet Standards is available in Section 2 of RFC 7841.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at <http://www.rfc-editor.org/info/rfc7895>.

Copyright Notice

Copyright (c) 2016 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
1.2. Tree Diagrams	3
2. YANG Module Library	4
2.1. modules-state	4
2.1.1. modules-state/module-set-id	4
2.1.2. modules-state/module	5
2.2. YANG Library Module	5
3. IANA Considerations	11
3.1. YANG Module Registry	11
4. Security Considerations	11
5. References	12
5.1. Normative References	12
5.2. Informative References	12
Acknowledgements	13
Authors' Addresses	13

1. Introduction

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so it is important that clients be able to cache the YANG library contents and easily identify whether their cache is out of date.

YANG library information can be different on every server and can change at runtime or across a server reboot.

If the server implements multiple protocols to access the YANG-defined data, each such protocol has its own conceptual instantiation of the YANG library.

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 name of the YANG module.
- 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 zero-length 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 MUST 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].

The following terms are defined in [RFC6241]:

- o client
- o server

The following terms are defined in [YANG1.1]:

- o module
- o submodule

The following terms are used within this document:

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

1.2. 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. This module is defined using YANG version 1, but it supports the description of YANG modules written in any revision of YANG.

Following is the YANG Tree Diagram for the "ietf-yang-library" module:

```
+--ro modules-state
  +--ro module-set-id      string
  +--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-type  enumeration
    +--ro submodule* [name revision]
      +--ro name            yang:yang-identifier
      +--ro revision        union
      +--ro schema?         inet:uri
```

2.1. modules-state

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

2.1.1. modules-state/module-set-id

This mandatory leaf contains a unique implementation-specific identifier representing the current set of modules and submodules on a specific server. The value of this leaf MUST change whenever the set of modules and submodules in the YANG library changes. There is no requirement that the same set always results in the same "module-set-id" value.

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

If the value of this leaf changes, the server also generates a "yang-library-change" notification, with the new value of "module-set-id".

Note that for a NETCONF server that implements YANG 1.1 [YANG1.1], a change of the "module-set-id" value results in a new value for the :yang-library capability defined in [YANG1.1]. Thus, if such a server implements NETCONF notifications [RFC5277], and the notification "netconf-capability-change" [RFC6470], a "netconf-capability-change" notification is generated whenever the "module-set-id" changes.

2.1.2. modules-state/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 each revision of each YANG module that is used by the server. It is possible for multiple revisions of the same module to be imported, in addition to an entry for the revision that is implemented 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.

<CODE BEGINS> file "ietf-yang-library@2016-06-21.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: <<https://datatracker.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) 2016 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 7895; see the RFC itself for full legal notices.";

revision 2016-06-21 {**description**

"Initial revision.";

reference

"RFC 7895: YANG Module Library.";

}

```
/*
 * Typedefs
 */

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

/*
 * Groupings
 */

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.
        A zero-length 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 revision of 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, regardless of whether they are
      defined in the module or any included submodule.";
  }
  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.

      The deviation module MUST be present in the 'module'
      list, with the same name and revision values.
      The 'conformance-type' value will be 'implement' for
      the deviation module.";
    uses common-leafs;
  }
}
```



```

    }
    leaf conformance-type {
      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 version 1.1 modules, there is at most one
            module entry with conformance type 'implement' for a
            particular module name, since YANG 1.1 requires that,
            at most, one revision of a module is implemented.

            For YANG version 1 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.";
        }
      }
    }
    mandatory true;
    description
      "Indicates the type of conformance the server is claiming
      for the YANG module identified by this entry.";
  }
  list submodule {
    key "name revision";
    description
      "Each entry represents one submodule within the
      parent module.";
    uses common-leafs;
    uses schema-leaf;
  }
}

```

```
/*
 * Operational state data nodes
 */

container modules-state {
  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.";
  }

  uses module-list;
}

/*
 * Notifications
 */

notification yang-library-change {
  description
    "Generated when the set of modules and submodules supported
     by the server has changed.";
  leaf module-set-id {
    type leafref {
      path "/yanglib:modules-state/yanglib:module-set-id";
    }
    mandatory true;
    description
      "Contains the module-set-id value representing the
       set of modules and submodules supported at the server at
       the time the notification is generated.";
  }
}

}

<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 has been made.

URI: urn:ietf:params:xml:ns:yang:ietf-yang-library
Registrant Contact: The NETCONF 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
reference: RFC 7895

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]. The NETCONF access control model [RFC6536] provides the means to restrict access for particular NETCONF users to a pre-configured subset of all available NETCONF protocol operations and content.

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-state/module: The module list used in a server implementation may help an attacker identify the server capabilities and server implementations with known bugs. Although some of this information may be available to all users via the NETCONF <hello> message (or similar messages in other management protocols), this YANG module potentially exposes additional details that could be of some assistance to an attacker. 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. References

5.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<http://www.rfc-editor.org/info/rfc2119>>.
- [RFC3688] Mealling, M., "The IETF XML Registry", BCP 81, RFC 3688, DOI 10.17487/RFC3688, January 2004, <<http://www.rfc-editor.org/info/rfc3688>>.
- [RFC6020] Bjorklund, M., Ed., "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", RFC 6020, DOI 10.17487/RFC6020, October 2010, <<http://www.rfc-editor.org/info/rfc6020>>.
- [RFC6241] Enns, R., Ed., Bjorklund, M., Ed., Schoenwaelder, J., Ed., and A. Bierman, Ed., "Network Configuration Protocol (NETCONF)", RFC 6241, DOI 10.17487/RFC6241, June 2011, <<http://www.rfc-editor.org/info/rfc6241>>.
- [RFC6242] Wasserman, M., "Using the NETCONF Protocol over Secure Shell (SSH)", RFC 6242, DOI 10.17487/RFC6242, June 2011, <<http://www.rfc-editor.org/info/rfc6242>>.
- [RFC6536] Bierman, A. and M. Bjorklund, "Network Configuration Protocol (NETCONF) Access Control Model", RFC 6536, DOI 10.17487/RFC6536, March 2012, <<http://www.rfc-editor.org/info/rfc6536>>.
- [RFC6991] Schoenwaelder, J., Ed., "Common YANG Data Types", RFC 6991, DOI 10.17487/RFC6991, July 2013, <<http://www.rfc-editor.org/info/rfc6991>>.

5.2. Informative References

- [RFC5277] Chisholm, S. and H. Trevino, "NETCONF Event Notifications", RFC 5277, DOI 10.17487/RFC5277, July 2008, <<http://www.rfc-editor.org/info/rfc5277>>.

- [RFC6470] Bierman, A., "Network Configuration Protocol (NETCONF) Base Notifications", RFC 6470, DOI 10.17487/RFC6470, February 2012, <<http://www.rfc-editor.org/info/rfc6470>>.
- [YANG1.1] Bjorklund, M., "The YANG 1.1 Data Modeling Language", Work in Progress, draft-ietf-netmod-rfc6020bis-12, April 2016.

Acknowledgements

Contributions to this material by Andy Bierman are based upon work supported by the Space & Terrestrial Communications Directorate (S&TCD) under Contract No. W15P7T-13-C-A616. Any opinions, findings, 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).

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