Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: May 3, 2018

A. Bierman
YumaWorks
M. Bjorklund
Tail-f Systems
K. Watsen
Juniper Networks
October 30, 2017

# YANG Data Extensions
# draft-bierman-netmod-yang-data-ext-01

## Abstract

This document describes YANG mechanisms for defining abstract data structures with YANG. It is intended to replace and extend the "yang-data" extension statement defined in RFC 8040.

## 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 May 3, 2018.

## Copyright Notice

Table of Contents



## 1. Introduction

There is a need for standard mechanisms to allow the definition of abstract data that is not intended to be implemented as configuration or operational state. The "yang-data" extension statement from RFC 8040 [RFC8040] is defined for this purpose, however it is limited in its functionality.

The intended use of the "yang-data" extension is to model all or part of a protocol message, such as the "errors" definition in ietf-restconf.yang [RFC8040], or the contents of a file. However, protocols are often layered such that the header or payload portions of the message can be extended by external documents. The YANG statements that model a protocol need to support this extensibility that is already found in that protocol.

This document defines a new YANG extension statement called "augment-yang-data", which allows abstract data structures to be augmented from external modules, similar to the existing YANG "augment" statement. Note that "augment" cannot be used to augment a yang data structure since a YANG compiler or other tool is not required to understand the "yang-data" extension.

The "yang-data" extension from [RFC8040] has been copied here and updated to be more flexible. There is no longer a requirement for

the "yang-data" statement to result in exactly one container object. There is no longer an assumption that a yang data structure can only be used as a top-level abstraction, instead of nested within some other data structure.

## 1.1. 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].

The following terms are used within this document:

- yang data structure: A data structure defined with the "yang-data" statement.

### 1.1.1. NMDA

The following terms are defined in the Network Management Datastore Architecture (NMDA) [I-D.ietf-netmod-revised-datastores]. and are not redefined here:

- configuration
- operational state

### 1.1.2. YANG

The following terms are defined in [RFC7950]:

- absolute-schema-nodeid
- container
- data definition statement
- data node
- leaf
- leaf-list
- list

## 2. Definitions

### 2.1. Restrictions on Conceptual YANG Data

This document places restrictions on the "yang-data" external statements that can be used with the "yang-data" and "augment-yang-data" extensions. The conceptual data definitions are considered to be in the same identifier namespace as defined in section 6.2.1 of [RFC7950]. In particular, bullet 7:

> All leafs, leaf-lists, lists, containers, choices, rpcs, actions, notifications, anydatas, and anyxmls defined (directly or through a "uses" statement) within a parent node or at the top level of the module or its submodules share the same identifier namespace.

This means that conceptual data defined with the "yang-data" or "augment-yang-data" statements cannot have the same local-name as sibling nodes from regular YANG data definition statements or other "yang-data" or "augment-yang-data" statements.

This does not mean a yang data structure has to be used as a top-level protocol message or other top-level data structure. A yang data structure does not have to result in a single container.

### 2.2. YANG Data Extensions Module

The "yang-data-ext" module defines the "augment-yang-data" extension to augment conceptual data already defined with the "yang-data" extension. The RESTCONF "yang-data" extension has been moved to this document and updated.

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

<CODE BEGINS> file "yang-data-ext@2017-10-30.yang"

```
module yang-data-ext {
  // not assigning real module name and namespace unless
  // and until changed to a draft-ietf-netmod document
  namespace "urn:ietf:params:xml:ns:yang:yang-data-ext";
  prefix "yd";

  organization
    "IETF NETMOD (NETCONF Data Modeling Language) Working Group";

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

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

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

Author: Kent Watsen
<mailto:kwatsen@juniper.net>";

```
  description
    "This module contains conceptual YANG specifications
     for defining abstract 'yang-data' data structures.

     Copyright (c) 2017 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).";

  revision 2017-10-30 {
    description
      "Initial revision.";
    reference
      "RFC XXXX: YANG Data Extensions.";
  }


  extension yang-data {
    argument name {
      yin-element true;
    }
    description
      "This extension is used to specify a YANG data template which
       represents conceptual data defined in YANG. It is
       intended to describe hierarchical data independent of
       protocol context or specific message encoding format.
       Data definition statements within a yang-data extension
       specify the generic syntax for the specific YANG data
       template, whose name is the argument of the yang-data
       extension statement.

       Note that this extension does not define a media-type.
       A specification using this extension MUST specify the
```

message encoding rules, including the content media type.

The mandatory 'name' parameter value identifies the YANG data template that is being defined. It contains the template name. This parameter is only used for readability purposes. There are no mechanisms to reuse yang-data by its template name value.

This extension is ignored unless it appears as a top-level statement. It MUST contain data definition statements that result in a set of data definition statements.

If the yang data template is intended to be used as a top-level structure, then the yang data template needs to result in a single container, so an instance of the YANG data template can thus be translated into an XML instance document, whose top-level element corresponds to the top-level container.

The module name and namespace value for the YANG module using the extension statement is assigned to each of the data definition statements resulting from the yang data template. The name of each data definition statement resulting from a yang data template is assigned to a top-level identifier name in the data node identifier namespace, according to RFC 7950, section 6.2.1.

The sub-statements of this extension MUST follow the 'data-def-stmt' rule in the YANG ABNF.

The XPath document root is the extension statement itself, such that the child nodes of the document root are represented by the data-def-stmt sub-statements within this extension. This conceptual document is the context for the following YANG statements:

- must-stmt
- when-stmt
- path-stmt
- min-elements-stmt
- max-elements-stmt
- mandatory-stmt
- unique-stmt
- ordered-by
- instance-identifier data type

The following data-def-stmt sub-statements are constrained when used within a yang-data-resource extension statement.

```
          - The list-stmt is not required to have a key-stmt defined.
          - The if-feature-stmt is ignored if present.
          - The config-stmt is ignored if present.
          - The available identity values for any 'identityref'
            leaf or leaf-list nodes is limited to the module
            containing this extension statement, and the modules
            imported into that module.
        ";
    }


    extension augment-yang-data {
      argument path {
        yin-element true;
      }
      description
        "This extension is used to specify an augmentation to
         conceptual data defined with the 'yang-data' statement.
         It is intended to describe hierarchical data independent
         of protocol context or specific message encoding format.

         This statement has almost the same structure as the
         'augment-stmt'. Data definition statements within this
         statement specify the semantics and generic syntax for the
         additional data to be added to the specific YANG data template,
         identified by the 'path' argument.

         The mandatory 'path' parameter value identifies the YANG
         conceptual data node that is being augmented, represented
         as an absolute-schema-nodeid string.

         This extension is ignored unless it appears as a top-level
         statement. The sub-statements of this extension MUST follow
         the 'data-def-stmt' rule in the YANG ABNF.

         The module name and namespace value for the YANG module using
         the extension statement is assigned to instance document data
         conforming to the data definition statements within
         this extension.

         The XPath document root is the augmented extension statement
         itself, such that the child nodes of the document root are
         represented by the data-def-stmt sub-statements within
         the augmented yang-data statement.

         The context node of the augment-yang-data statement is derived
         in the same way as the 'augment' statement, as defined in
         section 6.4.1 of [RFC7950]. This conceptual node is
```

considered the context node for the following YANG statements:

- must-stmt
- when-stmt
- path-stmt
- min-elements-stmt
- max-elements-stmt
- mandatory-stmt
- unique-stmt
- ordered-by
- instance-identifier data type

The following data-def-stmt sub-statements are constrained when used within a augment-yang-data extension statement.

- The list-stmt is not required to have a key-stmt defined.
- The if-feature-stmt is ignored if present.
- The config-stmt is ignored if present.
- The available identity values for any 'identityref' leaf or leaf-list nodes is limited to the module containing this extension statement, and the modules imported into that module.

Example:

```
   foo.yang {
      import ietf-restconf { prefix rc; }

      rc:yang-data foo-data {
        container foo-con { }
      }
   }

   bar.yang {
      import yang-data-ext { prefix yd; }
      import foo { prefix foo; }

      yd:augment-yang-data /foo:foo-con {
        leaf add-leaf1 { type int32; }
        leaf add-leaf2 { type string; }
      }
   }
";
  }

}
```

<CODE ENDS>

## 3. IANA Considerations

### 3.1. YANG Module Registry

TBD

## 4. Security Considerations

This document defines YANG extensions that are used to define conceptual YANG data. It does not introduce any new vulnerabilities beyond those specified in YANG 1.1 [RFC7950].

## 5. Normative References

[I-D.ietf-netmod-revised-datastores]
Bjorklund, M., Schoenwaelder, J., Shafer, P., Watsen, K., and R. Wilton, "Network Management Datastore Architecture", draft-ietf-netmod-revised-datastores-05 (work in progress), October 2017.

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

[RFC7950] Bjorklund, M., Ed., "The YANG 1.1 Data Modeling Language", RFC 7950, DOI 10.17487/RFC7950, August 2016, <http://www.rfc-editor.org/info/rfc7950>.

[RFC8040] Bierman, A., Bjorklund, M., and K. Watsen, "RESTCONF Protocol", RFC 8040, DOI 10.17487/RFC8040, January 2017, <http://www.rfc-editor.org/info/rfc8040>.

## Appendix A. Change Log

### A.1. v00 to v01

- Added Martin and Kent as authors
- Cloned and updated yang-data from RFC 8040
- Added text to clarify that yang-data does not have to result in a single container

## Appendix B. Open Issues

### B.1. uses-yang-data

Is there a need for a separate grouping and uses mechanism for yang-data? Currently only real grouping-stmt and uses-stmt are used.

### B.2. error-info

Is there a need for a special-purpose extension to define yang-data for the contents of the <error-info> node in NETCONF <rpc-error> and RESTCONF <errors> responses? This node is defined with anyxml so there is no way for a YANG tool to use real schema nodes, based on the RPC operation being requested or the error-app-tag that is being returned.

## 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