

Network Working Group  
Internet-Draft  
Intended status: Standards Track  
Expires: December 15, 2018

B. Lengyel  
Ericsson  
B. Claise  
Cisco Systems, Inc.  
June 13, 2018

**YANG Instance Data Files and their use for Documenting Server  
Capabilities  
draft-lengyel-netmod-yang-instance-data-01**

**Abstract**

This document specifies a standard file format for YANG instance data, that is data that could be stored in a datastore and whose syntax and semantics is defined by YANG models. Instance data files can be used to provide information that is defined in design time. There is a need to document Server capabilities (which are often specified in design time), which should be done using instance data files.

**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 <https://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 December 15, 2018.

**Copyright Notice**

Copyright (c) 2018 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 (<https://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.

## **1. Terminology**

**Design time:** A time during which a YANG model and the implementation behind it is created. Sometimes in other documents this period is divided into design and implementation time.

**Instance Data Set:** A named set of data items that can be used as instance data in a YANG data tree.

**Instance Data File:** A file containing an instance data set formatted according to the rules described in this document.

## **2. Introduction**

A YANG server has a number of server-capabilities that can be retrieved from the server using protocols like NETCONF or RESTCONF. YANG server capabilities include

- o data defined in ietf-yang-library (YANG modules, submodules, features, deviations, schema-mounts)
- o datastores supported
- o alarms supported ([draft-vallin-ccamp-alarm-module](#))
- o data nodes, subtrees that support or do not support on-change notifications ([draft-ietf-netconf-yang-push](#))
- o netconf-capabilities
- o etc.

While it is good practice to allow a client to query these capabilities from the live YANG server, that is often not enough. Many of these server-capabilities are relatively stable. They may change

1. only at upgrade, or
2. rarely (e.g. due to licensing), or
3. more frequently



Most capabilities belong to type 1), some to type 2) and a relatively small set to type 3). Many network nodes only have type 1) or type 1+2) capabilities. Stable capabilities are usually defined by a vendor in design time, before the product is released. While these capabilities can be retrieved from the live server in run-time, there is a strong need to provide the same data already during design time. (Often only a part of all the server capabilities can be made available.)

Often when a network node is released an associated NMS (network management system) is also released with it. The NMS depends on the capabilities of the YANG server. During NMS implementation information about server capabilities is needed. If the information is not available early in some off-line document, but only as instance data from the live network node, the NMS implementation will be delayed, because it has to wait for the network node to be ready. Also assuming that all NMS implementors will have a correctly configured network node available to retrieve data from, is a very expensive proposition. (An NMS may handle dozens of node types.)

Beside NMS implementors, system integrators and many others also need the same information early. Examples could be model driven testing, generating documentation, etc.

This document specifies a file format for YANG instance data and proposes to use it to provide server capability information, allowing vendors to specify capabilities early, in design time.

The same instance data file format can be used for other purposes, like providing initial data for any YANG module. E.g. a basic set of access control groups can be provided either by a device vendor or an operator using the network device.

## **2.1. Data Life cycle**

Data defined or documented in YANG Instance Data Sets may be used for preloading a YANG server with this data, but the server may populate the data without using the actual file in which case the Instance Data File is only used as documentation.

While such data will usually not change, data documented by Instance Data Files MAY be changed by the YANG server itself or by management operations. It is out of scope for this document to specify a method to prevent this.

Notifications about the change of data documented by Instance Data Sets may be supplied by e.g. the Yang-Push mechanism, but it is out of scope for this document.



### **2.2. Use Case 1: Early Documentation of Server Capabilities**

An operator wants to integrate his own in-house built management system with the network node from ACME Systems. The management integration must be ready by the time the first AcmeRouter 9000 is installed in the network. To do the integration the operator needs the the list of supported YANG modules and features. While this list could be read from the ietf-yang-library via Netconf, in order to allow time for developing the management integration, the operator demands this information early. The operator will value that this information is available in a standard format, that is actually the same format that can be read later from the node via Netconf.

### **2.3. Use Case 2: Preloading Data**

Defining Access control data is a complex task. To help with this, the device vendor pre-defines some of the data. Among others a set of default groups (/nacm:nacm/nacm:groups) e.g. "read-only", "operator", "sys-admin" and rules for these groups to access specific parts of common models could be defined. The operator will often use these default groups, but is also free to completely remove them and define his own set of groups.

## **3. Instance Data File Format**

Two standard formats to represent YANG Instance Data are specified based on the XML and JSON encoding. The XML format is defined in [[RFC7950](#)] while the JSON format is defined in [[RFC7951](#)]. Later as other YANG encodings (e.g. CBOR) are defined further Instance Data formats may be specified.

For both formats data is placed in a top level auxiliary container named "instance-data-set". The purpose of the container, which is not part of the real data itself, is to carry meta-data for the complete instance-data-set.

The XML format SHALL follow the format returned for a NETCONF GET operation. The <instance-data> anydata, which is not part of the real data itself, SHALL contain all data that would be inside the <data> wrapper element. XML attributes SHOULD NOT be used, however if a SW receiving a YANG instance data file encounters XML attributes unknown to it, it MUST ignore them, allowing them to be used later for other purposes.

The JSON format SHALL follow the format of the reply returned for a RESTCONF GET request directed at the datastore resource: {+restconf}/data. ETags and Timestamps SHOULD NOT be included, but if present SHOULD be ignored.



A YANG Instance data file MUST contain a single instance data set. Instance data MUST conform to the corresponding YANG Modules. Default values SHOULD NOT but MAY be included. Config=true and config=false data MAY be mixed in the instance data file. Instance data files MAY contain partial data sets. This means mandatory, min-elements or require-instance=true constrains MAY be violated.

Meta data, information about the data set itself SHALL be included in the instance data file. This data will be children of the top level instance-data-set container as defined in the ietf-instance-data YANG module. Meta data SHALL include:

- o Name of the instance data set

Meta data SHOULD include:

- o Revision date of the instance data set (later a semantic version MAY also be included)
- o Description of the instance data set. The description SHOULD contain information whether and how the data can change during the lifetime of the network element.

```
<instance-data-set xmlns=
  "urn:ietf:params:xml:ns:yang:ietf-yang-instance-data"
  <name>acme-router-modules</name>
  <revision>2108-01-25</revision>
  <description>Defines the minimal set of modules that any acme-router
    will contain. These modules will always be present.</description>
  <contact>info@acme.com</contact>
  <instance-data>
    <yang-library xmlns="urn:ietf:params:xml:ns:yang:ietf-yang-library">
      <module-set>
        <name>basic</name>
        <module>
          <name>ietf-system</>
          <revision>2014-08-06</revision>
          <!-- description "A later revision may be used."; -->
          <namespace>urn:ietf:params:xml:ns:yang:ietf-system</namespace>
          <feature>authentication</feature>
          <feature>radius-authentication</feature>
        </module>
      </module-set>
    </yang-library>
  </instance-data>
</instance-data-set>
```

Figure 1: XML Instance Data File example





```
{
  "ietf-yang-instance-data:instance-data-set": {
    "name": "acme-router-modules",
    "revision": "2108-01-25",
    "contact": "info@acme.com",
    "description":
      "Defines the set of modules that an acme-router will contain.",
    "instance-data": {
      "ietf-yang-library:yang-library": {
        "module-set": [
          "name": "basic",
          "module": [
            {
              "name": "ietf-system",
              "revision": "2014-08-06",
              "namespace": "urn:ietf:params:xml:ns:yang:ietf-system",
              "feature": ["authentication", "radius-authentication"]
            }
          ]
        ]
      }
    ]
  }
}
```

Figure 2: JSON Instance Data File example

#### 4. Provide Information about Server Capabilities

YANG instance data files SHOULD be used to provide design time information about server capabilities. The content of the instance data file SHOULD describe the capabilities of the server, however there is no general guarantee that the capabilities will not change over time. Whether capabilities change and if so, when and how SHOULD be described either in the instance data file description statements or some other implementation specific manner. If the information about server capabilities retrieved in run-time via Netconf/Restconf and the YANG Instance Data File for the same YANG module are in conflict, the run-time data MUST be considered authoritative.

The set of server capabilities to be documented will be defined by other standards and specifications, and is out of scope for this document. Whether and how the instance data files are used by SW implementing a YANG server is out of scope for this specification.



## 5. YANG Model

```
<CODE BEGINS> file "ietf-yang-instance-data.yang"

module ietf-yang-instance-data {
  yang-version 1.1;
  namespace
    "urn:ietf:params:xml:ns:yang:ietf-yang-instance-data";
  prefix ida ;

  import ietf-yang-data-ext { prefix yd; }

  organization "IETF NETMOD Working Group";
  contact
    "WG Web:  <https://datatracker.ietf.org/wg/netmod/>
    WG List:  <mailto:netmod@ietf.org>

    Author:   Balazs Lengyel
              <mailto:balazs.lengyel@ericsson.com>";

  description "The module defines the structure and content of YANG
    Instance Data Sets.";

  revision 2018-06-04 {
    description "Initial revision.";
    reference "RFC XXXX: YANG Instance Data";
  }

  yd:yang-data instance-data-format {
    container instance-data-set {
      description "Auxiliary container to carry meta-data for
        the complete instance data set.";

      leaf name {
        type string;
        mandatory true;
        description "Name of a YANG instance data set.";
      }

      leaf description { type string; }

      leaf contact {
        type string;
        description "Contains the same information the contact
          statement carries for a YANG module.";
      }

      leaf organization {
```



```
    type string;
    description "Contains the same information the
        organization statement carries for a YANG module.";
}

list revision {
    key date;
    description "An instance-data-set SHOULD have at least
        one revision entry. For every published
        editorial change, a new one SHOULD be added in front
        of the revisions sequence so that all revisions are
        in reverse chronological order.";

    leaf date {
        type string {
            pattern '\d{4}-\d{2}-\d{2}';
        }
        description "Specifies the data the revision
            was last modified.";
    }

    leaf description { type string; }
}

anydata instance-data {
    mandatory true;
    description "Contains the real instance data.
        The data MUST conform to the relevant YANG Modules.";
}
}
}
```

<CODE ENDS>

## **6. Security Considerations**

Depending on the nature of the instance data, instance data files MAY need to be handled in a secure way. The same type of handling should be applied, that would be needed for the result of a <get> operation returning the same data.

## **7. IANA Considerations**

To be completed, all the usual requests for a new YANG module



## 8. References

### 8.1. Normative References

- [I-D.ietf-netmod-yang-data-ext]  
Bierman, A., Bjorklund, M., and K. Watsen, "YANG Data Extensions", [draft-ietf-netmod-yang-data-ext-01](#) (work in progress), March 2018.
- [RFC7950] Bjorklund, M., Ed., "The YANG 1.1 Data Modeling Language", [RFC 7950](#), DOI 10.17487/RFC7950, August 2016, <<https://www.rfc-editor.org/info/rfc7950>>.
- [RFC7951] Lhotka, L., "JSON Encoding of Data Modeled with YANG", [RFC 7951](#), DOI 10.17487/RFC7951, August 2016, <<https://www.rfc-editor.org/info/rfc7951>>.

### 8.2. Informative References

- [I-D.ietf-netconf-rfc7895bis]  
Bierman, A., Bjorklund, M., Schoenwaelder, J., Watsen, K., and R. Wilton, "YANG Library", [draft-ietf-netconf-rfc7895bis-06](#) (work in progress), April 2018.
- [I-D.ietf-netconf-yang-push]  
Clemm, A., Voit, E., Prieto, A., Tripathy, A., Nilsen-Nygaard, E., Bierman, A., and B. Lengyel, "YANG Datastore Subscription", [draft-ietf-netconf-yang-push-16](#) (work in progress), May 2018.
- [I-D.vallin-ccamp-alarm-module]  
Vallin, S. and M. Bjorklund, "YANG Alarm Module", [draft-vallin-ccamp-alarm-module-01](#) (work in progress), October 2017.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in [RFC 2119](#) Key Words", [BCP 14](#), [RFC 8174](#), DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.

## Appendix A. Changes between revisions





- o Redefined using yang-data-ext
- o Moved meta data into ordinary leafs/leaf-lists

#### Authors' Addresses

Balazs Lengyel  
Ericsson  
xxx  
1117 Budapest  
Hungary

Phone: +36-70-330-7909  
Email: [balazs.lengyel@ericsson.com](mailto:balazs.lengyel@ericsson.com)

Benoit Claise  
Cisco Systems, Inc.  
De Kleetlaan 6a b1  
1831 Diegem  
Belgium

Phone: +32 2 704 5622  
Email: [bclaise@cisco.com](mailto:bclaise@cisco.com)

