

NETMOD Working Group  
Internet-Draft  
Intended status: Standards Track  
Expires: June 2, 2016

L. Lhotka  
CZ.NIC  
November 30, 2015

YANG Schema Dispatching Language  
draft-lhotka-netmod-ysdl-00

## Abstract

This document defines YANG Schema Dispatching Language (YSDL). It is a meta-schema language that allows for combining YANG modules into any number of schemas, and arranging these schemas in a hierarchical structure.

## 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 2, 2016.

## Copyright Notice

Copyright (c) 2015 IETFTrust 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.

Internet-Draft

i-d-abbrev

November 2015

## Table of Contents

<a href="#">1.</a>	Introduction . . . . .	<a href="#">2</a>
<a href="#">2.</a>	Terminology and Notation . . . . .	<a href="#">2</a>
<a href="#">3.</a>	YANG Schema Dispatching Language . . . . .	<a href="#">4</a>
<a href="#">3.1.</a>	YSDL Schemas . . . . .	<a href="#">4</a>
<a href="#">3.2.</a>	Validating Data with YSDL and YANG . . . . .	<a href="#">5</a>
<a href="#">4.</a>	YSDL YANG Module . . . . .	<a href="#">5</a>
<a href="#">5.</a>	Example YSDL Meta-Schema . . . . .	<a href="#">8</a>
<a href="#">6.</a>	IANA Considerations . . . . .	<a href="#">11</a>
<a href="#">7.</a>	Security Considerations . . . . .	<a href="#">12</a>
<a href="#">8.</a>	Acknowledgements . . . . .	<a href="#">12</a>
<a href="#">9.</a>	References . . . . .	<a href="#">12</a>
<a href="#">9.1.</a>	Normative References . . . . .	<a href="#">12</a>
<a href="#">9.2.</a>	Informative References . . . . .	<a href="#">13</a>
	Author's Address . . . . .	<a href="#">13</a>

[1.](#) Introduction

YANG data modelling language [[I-D.ietf-netmod-rfc6020bis](#)] assumes that data trees (configuration and state data) defined in YANG modules start at the common global root. This means that the global path of every data node is fixed as soon as the data node is defined (for a node defined in a grouping, the path is fixed when the grouping is used).

Recent data modeling efforts, such as [[I-D.rtgyangdt-rtgwg-device-model](#)] indicate that this arrangement may be too restrictive. A typical use case is a data model that was designed for a stand-alone device but later needs to be reused for logical or virtual devices, and the original data model has to be grafted as a submodel in a top-level schema describing the organization of logical/virtual devices.

This document proposes a new minilanguage named YANG Schema Dispatching Language (YSDL) as a solution to this problem. It is a meta-schema language that allows for defining multiple schemas comprising one or more YANG modules, and arrange the schemas in a hierarchical structure.

As a useful side-effect, it is also possible to use YSDL schemas to specify a data model for an "anydata" node at run time.

## [2.](#) Terminology and Notation

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 defined in [[I-D.ietf-netmod-rfc6020bis](#)]:

- o action,
- o anydata,
- o augment,
- o container,
- o data node,
- o data tree,
- o identity,
- o instance identifier,
- o leaf,
- o leaf-list,
- o list,
- o module,
- o notification,
- o RPC operation,
- o schema node.

The term "server" denotes a server of a network management protocol such as NETCONF [[RFC6241](#)] or RESTCONF [[I-D.ietf-netconf-restconf](#)].

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 Curly braces "{" and "}" contain names of optional features that make the corresponding node conditional.
- o Abbreviations before data node names: "rw" means configuration (read-write), "ro" state data (read-only), "-x" RPC operations, and "-n" notifications.

- o Symbols after data node names: "?" means an optional node, "!" a container with presence, and "\*" denotes a "list" or "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.

### 3. YANG Schema Dispatching Language

The YSDL syntax is defined in [Section 4](#) using YANG in the role of a schema language. YSDL meta-schemas can be serialized in any encoding supported by YANG, such as XML [[I-D.ietf-netmod-rfc6020bis](#)] or JSON [[I-D.ietf-netmod-yang-json](#)].

YSDL meta-schemas are intended to be used in conjunction with YANG Library [[I-D.ietf-netconf-yang-library](#)]: names and revisions of YANG modules appearing in a meta-schema SHALL be resolved from YANG Library data, and YANG Library specifications of supported features and deviations MUST also be taken into account.

A YSDL meta-schema only affects the main data tree (configuration and state data), including actions. RPC operations and notifications are handled exactly as specified in [[I-D.ietf-netmod-rfc6020bis](#)].

YANG tree diagram for a YSDL meta-schema is as follows:

```
module: ietf-ysdl
```

```

+---rw schemas
  +---rw top-schema?  schema-ref
  +---rw schema* [name]
    +---rw name          string
    +---rw yang-modules* yang:yang-identifier
    +---rw subschema* [root]
      +---rw root        schema-node-path
      +---rw schemas*    schema-ref

```

### [3.1.](#) YSDL Schemas

A YSDL meta-schema defines one or more schemas, each comprising one or more YANG modules and zero or more subschemas. The same YANG module can be part of multiple schemas.

Exactly one of the schemas MUST be the top-level schema which is identified by the "top-schema" leaf. The top-level schema defines the uppermost structure of the entire data model, and its root always coincides with the global root.

Any schema can define other schemas as its subschema. For each subschema, the root node has to be specified. Any "container", "list", "case" or "anydata" schema node contained in one of the parent schema modules can act as the root node.

Note that the traditional arrangement of YANG modules in a data model can be achieved with a YSDL meta-schema in which all modules are listed in the top-level schema, and no subschemas are defined.

### [3.2.](#) Validating Data with YSDL and YANG

The schema tree of a subschema is conceptually attached to the parent schema tree as a subtree of the root node. In other words, the first data node that is either the subschema root or its ancestor coincides with the conceptual root of the subschema data tree.

All paths (in leafrefs, instance-identifiers, XPath expressions and target nodes of augments) are interpreted within the subschema data or schema tree. Therefore, they cannot refer to nodes in the ancestor schemas or other subschemas.

## [4.](#) YSDL YANG Module

The "ietf-ysdl" module below defines the YSDL meta-schema language using YANG. However, it is not meant to be implemented as datastore contents by a server.

```
<CODE BEGINS> file "ietf-ysdl@2015-11-30.yang"

module ietf-ysdl {

    namespace "urn:ietf:params:xml:ns:yang:ietf-ysdl";

    prefix "ysdl";

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

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

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

        WG Chair: Thomas Nadeau
                  <mailto:tnadeau@lucidvision.com>
```

Lhotka

Expires June 2, 2016

[Page 5]

---

Internet-Draft

i-d-abbrev

November 2015

WG Chair: Juergen Schoenwaelder  
<mailto:j.schoenwaelder@jacobs-university.de>

WG Chair: Kent Watsen  
<mailto:kwatsen@juniper.net>

Editor: Ladislav Lhotka  
<mailto:lhotka@nic.cz>;

#### description

"This module defines YANG Schema Dispatching Language (YSDL) - a meta-schema language for YANG-based data models.

A YSDL schema allows for embedding a YANG schema as a subschema in a specific location of another module.

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>).

The key words 'MUST', 'MUST NOT', 'REQUIRED', 'SHALL', 'SHALL NOT', 'SHOULD', 'SHOULD NOT', 'RECOMMENDED', 'MAY', and 'OPTIONAL' in the module text are to be interpreted as described in [RFC 2119](#) (<http://tools.ietf.org/html/rfc2119>).

This version of this YANG module is part of RFC XXXX (<http://tools.ietf.org/html/rfcXXXX>); see the RFC itself for full legal notices.";

```
revision 2015-11-30 {
  description
    "Initial revision.";
  reference
    "RFC XXXX: YANG Schema Dispatching Language";
}

/* Typedefs */

typedef schema-node-path {
  type string {
    pattern "[a-zA-Z_][a-zA-Z0-9\\-\\.]*:[a-zA-Z_][a-zA-Z0-9\\-\\.]*"
      + "(/([a-zA-Z_][a-zA-Z0-9\\-\\.]*"

```

```
      + "(:)?[a-zA-Z_][a-zA-Z0-9\\-\\.]*");
}
description
  "Intra-schema path to a YANG schema node.
```

The value is a sequence of schema node names separated by the slash character ('/'). The first (leftmost) component is prefixed with the name of the YANG module in which the schema

```

        node is defined. A subsequent component has the module prefix
        if and only if the preceding schema node is defined in another
        module.";
    }

typedef schema-ref {
    type leafref {
        path "/ysdl:schemas/ysdl:schema/ysdl:name";
    }
    description
        "Reference to a schema.";
}

/* YSDL */

container schemas {
    description
        "Container for YSDL schemas.";
    leaf top-schema {
        type schema-ref;
        description
            "Reference to the top-level schema.";
    }
    list schema {
        key "name";
        min-elements "1";
        description
            "Each entry is a named schema. Exactly one entry must be the
            top-level schema, other (optional) entries can be used as
            subschemas.";
        leaf name {
            type string;
            description
                "Name of the schema.";
        }
        leaf-list yang-modules {
            type yang:yang-identifier;
            min-elements "1";
            description
                "Each entry is the name of a YANG module contributing to

```

the schema.";



```

    }
    list subschema {
        key "root";
        description
            "Each entry is a subschema attached to the parent
            schema.";
        leaf root {
            type schema-node-path;
            description
                "Each entry specifies the relative (intra-schema) path to
                the root node where the subschema is attached. The
                schema node MUST be one of: container, list, case or
                anydata.

                The absolute path of the root node depends on the
                structure of schemas:

                - For the top-level schema, the root is always '/'.

                - For other schemas, the root path is formed by
                  concatenation of the corresponding root paths of all
                  ancestor schemas.";
        }
        leaf-list schemas {
            type schema-ref;
            description
                "References to schemas that comprise the subschema.";
        }
    }
}
}
}
}
}

```

<CODE ENDS>

## 5. Example YSDL Meta-Schema

The following YANG module will be used for the top-level schema. It is a simplified analogy of the "network-device" module from [\[I-D.rtgyangdt-rtgwg-device-model\]](#).

```
module example-device {  
    namespace "http://www.example.net/example-device";  
    prefix "exdev";  
    container device {  
        container logical-NEs {  
            list logical-NE {  
                key "name";  
                leaf name {  
                    type string;  
                }  
            }  
        }  
    }  
}
```

The YSDL meta-schema is as follows:

Internet-Draft

i-d-abbrev

November 2015

```
{
  "ietf-ysdl:schemas": {
    "top-schema": "device",
    "schema": [
      {
        "name": "device",
        "yang-modules": [
          "example-device"
        ],
        "subschema": [
          {
            "root": "example-device:device",
            "schemas": [
              "if-ip",
              "system"
            ]
          },
          {
            "root": "example-device:device/logical-NEs/logical-NE",
            "schemas": [
              "if-ip"
            ]
          }
        ]
      },
      {
        "name": "if-ip",
        "yang-modules": [
          "ietf-interfaces",
          "ietf-ip"
        ]
      },
      {
        "name": "system",
        "yang-modules": [
          "ietf-system"
        ]
      }
    ]
  }
}
```

```
}  
}
```

Apart from the top-level schema, "device", the meta-schema defines two other schemas:

- o "if-ip" schema consists of "ietf-interfaces" and "ietf-ip" modules;

- o "system" schema consist of "ietf-system" module.

Two subschemas of the top-level schema defined with the following root nodes:

- o "example-device:device",
- o "example-device:device/logical-NEs/logical-NE".

The former one uses both "if-ip" and "system" schemas whereas the latter uses only "system".

The entire schema tree consisting of the top-level schema and the two subschemas is shown schematically in the following tree diagram:

```
module: example-device  
  +--rw device  
    +--rw if:interfaces  
      | ...  
    +--ro if:interfaces-state  
      | ...  
    +--rw sys:system  
      | ...  
    +--rw sys:system-state  
    +--rw logical-NEs  
      +--rw logical-NE* [name]  
        +--rw name      string  
        +--rw if:interfaces  
          | ...  
        +--ro if:interfaces-state  
          | ...
```

## 6. IANA Considerations

RFC Editor: In this section, replace all occurrences of 'XXXX' with the actual RFC number and all occurrences of the revision date below with the date of RFC publication (and remove this note).

This document registers a URI in the "IETF XML registry" [[RFC3688](#)]. Following the format in [RFC 3688](#), the following registration has been made.

Lhotka

Expires June 2, 2016

[Page 11]

---

Internet-Draft

i-d-abbrev

November 2015

-----  
URI: urn:ietf:params:xml:ns:yang:ietf-ysdl

Registrant Contact: The NETMOD WG of the IETF.

XML: N/A, the requested URI is an XML namespace.  
-----

This document registers a YANG module in the "YANG Module Names" registry [[RFC6020](#)].

-----  
name: ietf-ysdl  
namespace: urn:ietf:params:xml:ns:yang:ietf-ysdl  
prefix: ysdL  
reference: RFC XXXX  
-----

## 7. Security Considerations

TBD

## 8. Acknowledgements

YSDL was loosely inspired by Namespace-based Validation Dispatching

Language (NVDL) [[ISO.19757-4](#)].

## [9.](#) References

### [9.1.](#) Normative References

- [I-D.ietf-netconf-yang-library]  
Bierman, A., Bjorklund, M., and K. Watsen, "YANG Module Library", [draft-ietf-netconf-yang-library-02](#) (work in progress), October 2015.
- [I-D.ietf-netmod-rfc6020bis]  
Bjorklund, M., "The YANG 1.1 Data Modeling Language", [draft-ietf-netmod-rfc6020bis-08](#) (work in progress), October 2015.
- [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>>.

Lhotka

Expires June 2, 2016

[Page 12]

---

Internet-Draft

i-d-abbrev

November 2015

- [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>>.

### [9.2.](#) Informative References

- [I-D.ietf-netconf-restconf]  
Bierman, A., Bjorklund, M., and K. Watsen, "RESTCONF Protocol", [draft-ietf-netconf-restconf-08](#) (work in progress), October 2015.
- [I-D.ietf-netmod-yang-json]  
Lhotka, L., "JSON Encoding of Data Modeled with YANG",

[draft-ietf-netmod-yang-json-06](#) (work in progress), October 2015.

[I-D.rtgyangdt-rtgwg-device-model]

Lindem, A., Berger, L., Bogdanovic, D., and C. Hopps, "Network Device YANG Organizational Model", [draft-rtgyangdt-rtgwg-device-model-01](#) (work in progress), September 2015.

[ISO.19757-4]

International Organization for Standardization, "Document Schema Definition Languages (DSDL) - Part 4: Namespace-based Validation Dispatching Language (NVDL)", ISO/IEC 19757-4, June 2006.

[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>>.

#### Author's Address

Ladislav Lhotka  
CZ.NIC

Email: [lhotka@nic.cz](mailto:lhotka@nic.cz)