

Internet Engineering Task Force
Internet-Draft
Intended status: Standards Track
Expires: June 22, 2019

M. Veillette, Ed.
Trilliant Networks Inc.
A. Pelov, Ed.
I. Petrov, Ed.
Acklio
December 19, 2018

**YANG Schema Item iDentifier (SID)
draft-ietf-core-sid-05**

Abstract

YANG Schema Item iDentifiers (SID) are globally unique 64-bit unsigned numbers used to identify YANG items. This document defines the semantics, the registration, and assignment processes of SIDs. To enable the implementation of these processes, this document also defines a file format used to persist and publish assigned SIDs.

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 June 22, 2019.

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.

Table of Contents

1.	Introduction	2
2.	Terminology and Notation	3
3.	".sid" file lifecycle	4
4.	".sid" file format	7
5.	Third party registries	11
6.	Security Considerations	11
7.	IANA Considerations	11
7.1.	Module registration	12
7.2.	"SID mega-range" registry	12
7.2.1.	"IANA SID Mega-Range" allocation	13
7.2.2.	"RFC SID range assignment" sub-registry	14
7.2.3.	"Specification SID range assignment" sub-registry	14
7.3.	"YANG module assignment" registry	15
8.	Acknowledgments	15
9.	References	16
9.1.	Normative References	16
9.2.	Informative References	16
Appendix A.	".sid" file example	17
	Authors' Addresses	26

1. Introduction

Some of the items defined in YANG [[RFC7950](#)] require the use of a unique identifier. In both NETCONF [[RFC6241](#)] and RESTCONF [[RFC8040](#)], these identifiers are implemented using names. To allow the implementation of data models defined in YANG in constrained devices and constrained networks, a more compact method to identify YANG items is required. This compact identifier, called SID, is encoded using a 64-bit unsigned integer. The following items are identified using SIDs:

- o identities
- o data nodes (Note: including those part of a YANG template as defined by the 'yang-data' extension.)
- o RPCs and associated input(s) and output(s)
- o actions and associated input(s) and output(s)
- o notifications and associated information
- o YANG modules, submodules and features

To minimize their size, SIDs are often represented as a difference between the current SID and a reference SID. Such difference is called "delta", shorthand for "delta-encoded SID". Conversion from SIDs to deltas and back to SIDs is a stateless process. Each protocol implementing deltas must unambiguously define the reference SID for each YANG item.

SIDs are globally unique numbers, a registration system is used in order to guarantee their uniqueness. SIDs are registered in blocks called "SID ranges".

Assignment of SIDs to YANG items can be automated, the recommended process to assign SIDs is as follows:

1. A tool extracts the different items defined for a specific YANG module.
2. The list of items is sorted in alphabetical order, 'namespace' in descending order, 'identifier' in ascending order. The 'namespace' and 'identifier' formats are described in the YANG module 'ietf-sid-file' defined in [Section 4](#).
3. SIDs are assigned sequentially from the entry point up to the size of the registered SID range. This approach is recommended to minimize the serialization overhead, especially when delta encoding is implemented.
4. If the number of items exceeds the SID range(s) allocated to a YANG module, an extra range is added for subsequent assignments.

SIDs are assigned permanently, items introduced by a new revision of a YANG module are added to the list of SIDs already assigned. This process can also be automated using the same method described above, only unassigned YANG items are processed at step #3.

[Section 3](#) provides more details about the registration process of YANG modules and associated SIDs. To enable the implementation of this registry, [Section 4](#) defines a standard file format used to store and publish SIDs.

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 [\[RFC7950\]](#):

- o action
- o feature
- o module
- o notification
- o RPC
- o schema node
- o schema tree
- o submodule

The following term is defined in [[RFC8040](#)]:

- o yang-data extension

This specification also makes use of the following terminology:

- o delta : Difference between the current SID and a reference SID. Each protocol that uses delta encoded SIDs MUST define how the reference SID is obtained.
- o item: A schema node, an identity, a module, a submodule or a feature defined using the YANG modeling language.
- o path: A path is a string that identifies a schema node within the schema tree. A path consists of the list of schema node identifier(s) separated by slashes ("/"). Schema node identifier(s) are always listed from the top-level schema node up to the targeted schema node. (e.g. "/ietf-system:system-state/clock/current-datetime")
- o YANG Schema Item iDentifier (SID): Unsigned integer used to identify different YANG items.

3. ".sid" file lifecycle

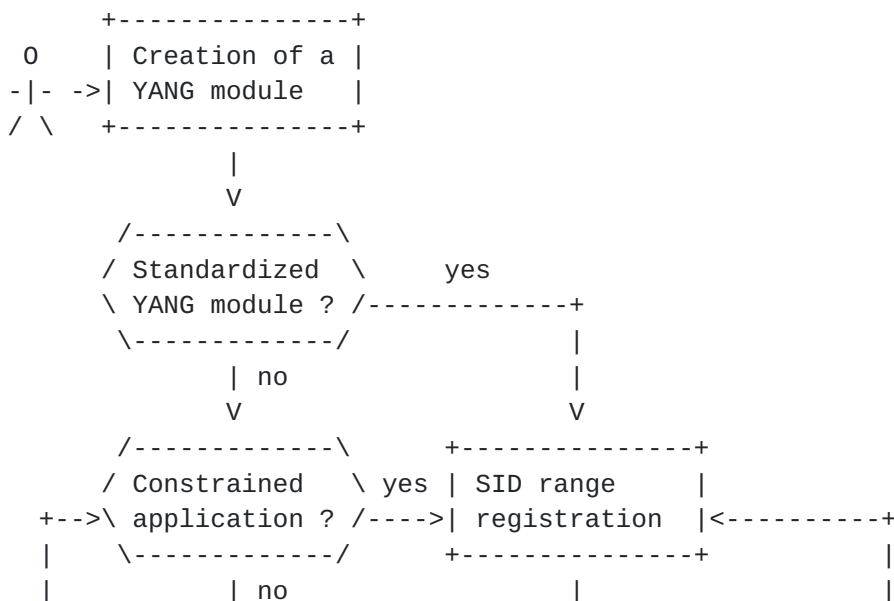
YANG is a language designed to model data accessed using one of the compatible protocols (e.g. NETCONF [[RFC6241](#)], RESCONF [[RFC8040](#)] and CoMI [[I-D.ietf-core-comi](#)]). A YANG module defines hierarchies of data, including configuration, state data, RPCs, actions and notifications.

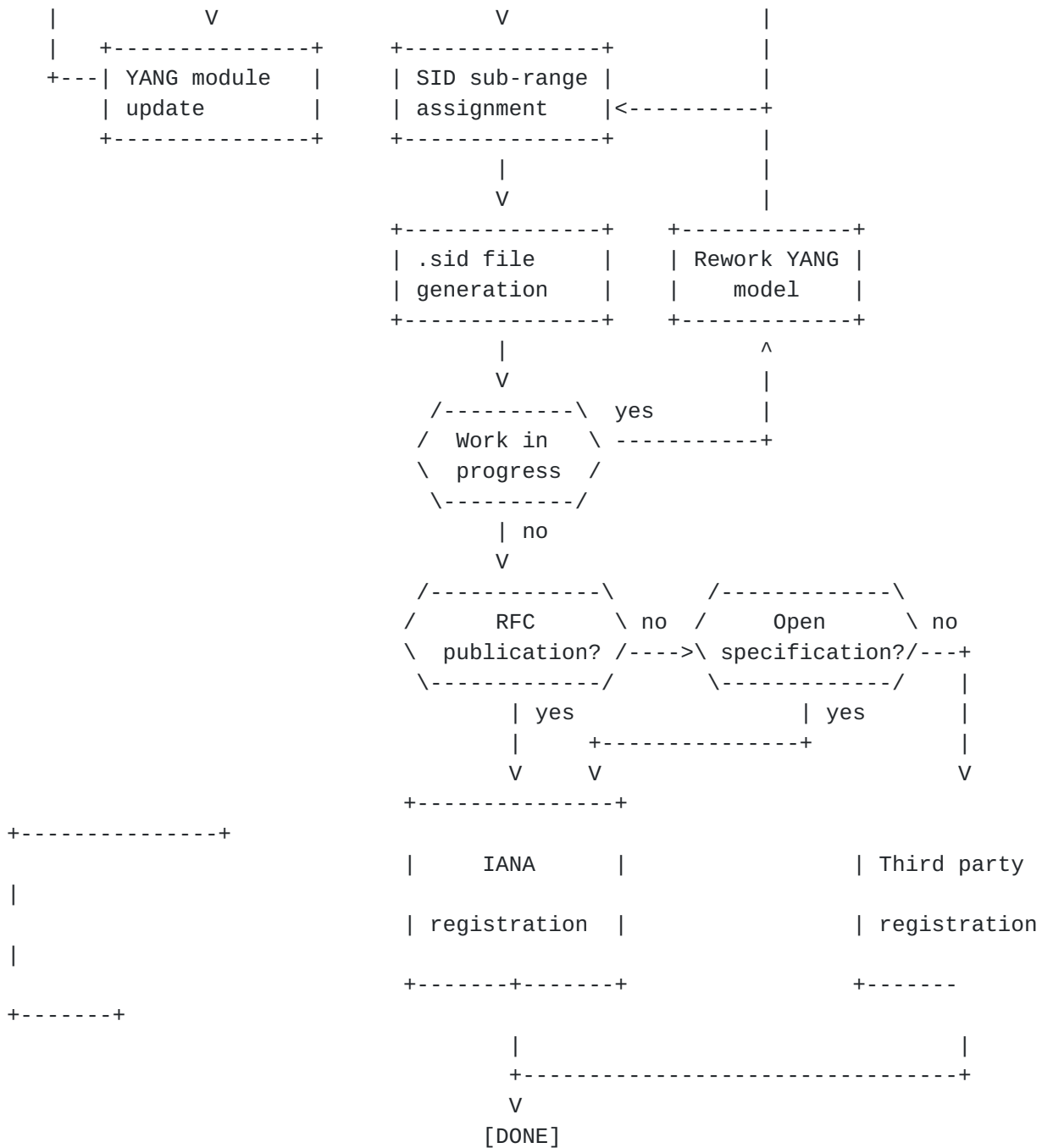
YANG modules are not necessarily created in the context of constrained applications. YANG modules can be implemented using NETCONF [RFC6241] or RESTCONF [RFC8040] without the need to assign SIDs.

As needed, authors of YANG modules can assign SIDs to their YANG modules. In order to do that, they should first obtain a SID range from a registry. It could be "RFC SID range assignment" sub-registry as defined in Section 7.2.2, the "Specification SID range assignment" sub-registry as defined in Section 7.2.3 or another one, depending on the particular case. The minimal information required for this would be a start SID number and a range size, but might include additional details depending on the registry policy, which is outside the scope of this document. Once a SID range is registered, the owner can use it to generate ".sid" file/s for his YANG module/s. It is recommended to leave some unallocated SIDs following the allocated range in each ".sid" file in order to allow better evolution of the YANG module in the future. Generation of ".sid" files SHOULD be performed using an automated tool. Note that ".sid" files can only be generated for YANG modules and not for submodules.

Registration of the .sid file associated to a YANG module is optional but recommended to promote interoperability between devices and to avoid duplicate allocation of SIDs to a single YANG module. Different registries might have different requirement for the registration and publication of the ".sid" files.

The following activity diagram summarizes the creation of a YANG module and its associated .sid file.





Each time a YANG module or one of its imported module(s) or included sub-module(s) is updated, the ".sid" file MAY need to be updated. This update SHOULD also be performed using an automated tool.

If a new revision requires more SIDs than initially allocated, a new SID range MUST be added to the 'assignment-ranges' as defined in [Section 4](#). These extra SIDs are used for subsequent assignments.

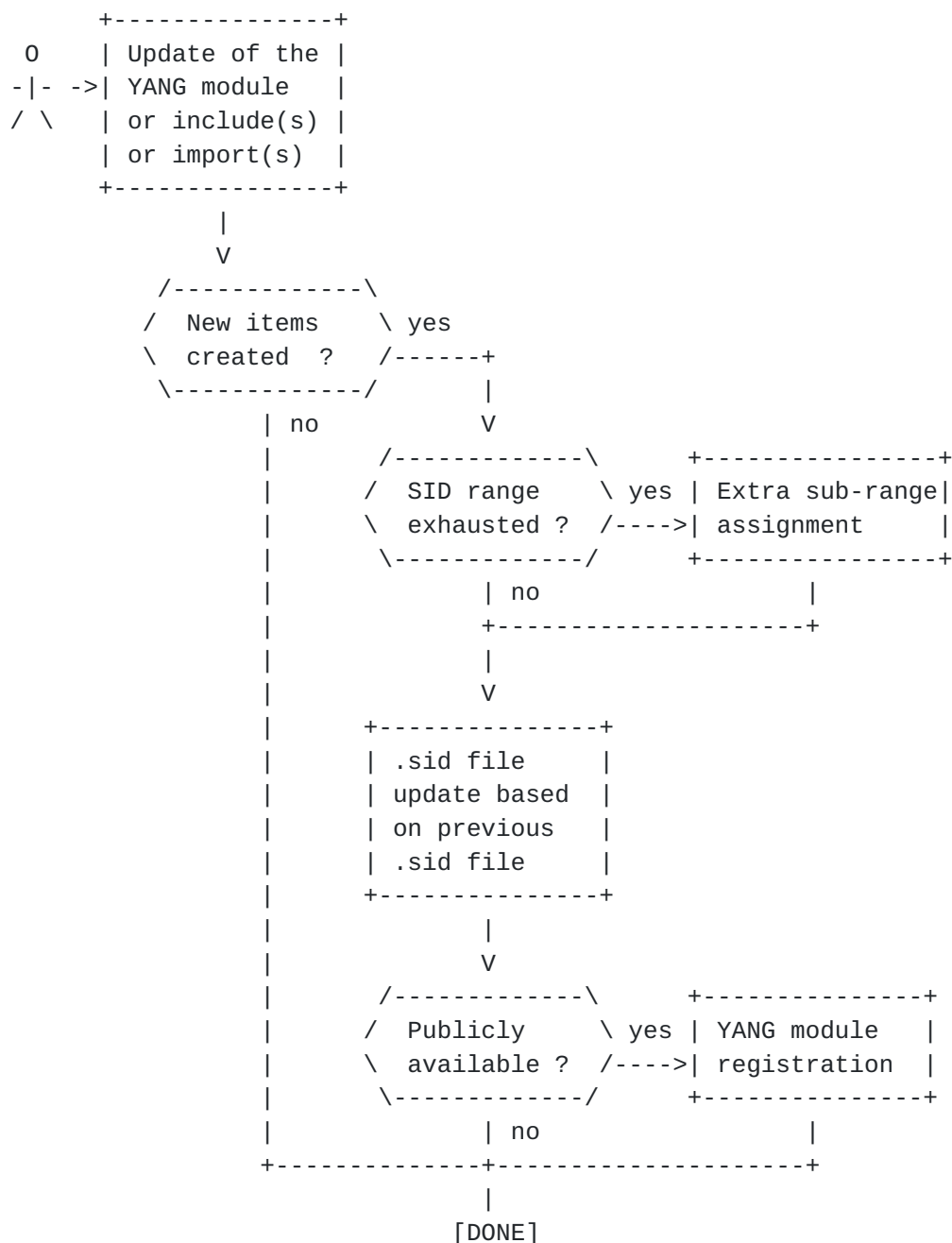
The following activity diagram summarizes the update of a YANG module

and its associated .sid file.

Veillette, et al.

Expires June 22, 2019

[Page 6]



4. ".sid" file format

".sid" files are used to persist and publish SIDs assigned to the different YANG items of a specific YANG module. The following YANG module defined the structure of this file, encoding is performed using the rules defined in [\[RFC7951\]](#).

```

<CODE BEGINS> file "ietf-sid-file@2017-11-26.yang"
module ietf-sid-file {

```



```
namespace "urn:ietf:params:xml:ns:yang:ietf-sid-file";
prefix sid;

import ietf-yang-types {
  prefix yang;
}

import ietf-comi {
  prefix comi;
}

organization
  "IETF Core Working Group";

contact
  "Michel Veillette
  <mailto:michel.veillette@trilliant.com>

  Andy Bierman
  <mailto:andy@yumaworks.com>

  Alexander Pelov
  <mailto:a@ackl.io>";

description
  "This module defines the structure of the .sid files.

  Each .sid file contains the mapping between the different
  string identifiers defined by a YANG module and a
  corresponding numeric value called SID.";

revision 2017-11-26 {
  description
    "Initial revision.";
  reference
    "[I-D.ietf-core-sid] YANG Schema Item iDentifier (SID)";
}

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

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
    "Identifies a schema-node path string for use in the
    SID registry. This string format follows the rules
    for an instance-identifier, as defined in RFC 7959,
    except that no predicates are allowed.

    This format is intended to support the YANG 1.1 ABNF
    for a schema node identifier, except module names
    are used instead of prefixes, as specified in RFC 7951.";
  reference
    "RFC 7950, The YANG 1.1 Data Modeling Language;
    Section 6.5: Schema Node Identifier;
    RFC 7951, JSON Encoding of YANG Data;
    Section 6.11: The instance-identifier type";
}

leaf module-name {
  type yang:yang-identifier;
  description
    "Name of the YANG module associated with this .sid file.";
}

leaf module-revision {
  type revision-identifier;
  description
    "Revision of the YANG module associated with this .sid file.
    This leaf is not present if no revision statement is
    defined in the YANG module.";
}

list assigment-ranges {
  key "entry-point";
  description
    "SID range(s) allocated to the YANG module identified by
    'module-name' and 'module-revision'.";

  leaf entry-point {
    type comi:sid;
    mandatory true;
    description
      "Lowest SID available for assignment.";
  }

  leaf size {
```



```
    type uint64;
    mandatory true;
    description
      "Number of SIDs available for assignment.";
  }
}

list items {
  key "namespace identifier";
  description
    "Each entry within this list defined the mapping between
    a YANG item string identifier and a SID. This list MUST
    include a mapping entry for each YANG item defined by
    the YANG module identified by 'module-name' and
    'module-revision'.";

  leaf namespace {
    type enumeration {
      enum module {
        value 0;
        description
          "All module and submodule names share the same
          global module identifier namespace.";
      }
      enum identity {
        value 1;
        description
          "All identity names defined in a module and its
          submodules share the same identity identifier
          namespace.";
      }
      enum feature {
        value 2;
        description
          "All feature names defined in a module and its
          submodules share the same feature identifier
          namespace.";
      }
      enum data {
        value 3;
        description
          "The namespace for all data nodes, as defined in YANG.";
      }
    }
  }
  description
    "Namespace of the YANG item for this mapping entry.";
}
```



```
leaf identifier {
  type union {
    type yang:yang-identifier;
    type schema-node-path;
  }
  description
    "String identifier of the YANG item for this mapping entry.

    If the corresponding 'namespace' field is 'module',
    'feature', or 'identity', then this field MUST
    contain a valid YANG identifier string.

    If the corresponding 'namespace' field is 'data',
    then this field MUST contain a valid schema node
    path.";
}

leaf sid {
  type comi:sid;
  mandatory true;
  description
    "SID assigned to the YANG item for this mapping entry.";
}
}
}
<CODE ENDS>
```

5. Third party registries

The organization and functioning of third party registries is outside the scope of the current document. The only limitations connected to those registries are listed in [Section 7.2](#).

6. Security Considerations

The security considerations of [\[RFC7049\]](#) and [\[RFC7950\]](#) apply.

This document defines a new type of identifier used to encode data models defined in YANG [\[RFC7950\]](#). As such, this identifier does not contribute to any new security issues in addition of those identified for the specific protocols or contexts for which it is used.

7. IANA Considerations

In this section are given specifications for an entry into the module registry and two new registries, a SID-range registry and a SID module registry.

7.1. Module registration

This document registers one YANG modules in the "YANG Module Names" registry [[RFC6020](#)]:

- o name: ietf-sid-file
- o namespace: urn:ietf:params:xml:ns:yang:ietf-sid-file
- o prefix: sid
- o reference: [[THISRFC]]

7.2. "SID mega-range" registry

The name of this registry is "SID mega-range". This registry is used to record the delegation of the management of a block of SIDs to third parties (e.g. SDO, registrar).

Each entry in this registry must include:

- o The entry point (first entry) of the registered SID range.
- o The size of the registered SID range.
- o The contact information of the requesting organization including:
 - o Organization name
 - o Primary contact name, email address, and phone number
 - o Secondary contact name, email address, and phone number

The initial entry in this registry is allocated to IANA:

```

+-----+-----+-----+
| Entry Point | Size   | Organization name |
+-----+-----+-----+
| 0           | 1000000 | IANA              |
+-----+-----+-----+

```

The IANA policies for future additions to this registry are "Hierarchical Allocation, Expert Review" [[RFC5226](#)]. Prior to a first allocation, the requesting organization must demonstrate a functional registry infrastructure. On subsequent allocation request(s), the organization must demonstrate the exhaustion of the prior range. These conditions need to be asserted by the assigned expert(s).

7.2.1. "IANA SID Mega-Range" allocation

The first million SIDs assigned to IANA is sub-divided as follow:

- o The range of 0 to 999 is reserved for future extensions. The IANA policy for this range is "IETF review" [[RFC5226](#)]. This range is reserved for a future uses and no sub-registries are currently defined for it.
- o The range of 1000 to 59,999 is reserved for YANG modules defined in RFCs. The IANA policy for future additions to this sub-registry is "RFC required" [[RFC5226](#)]. Allocation within this range requires publishing of the associated ".yang" and ".sid" files in the YANG module registry. The allocation within this range is done during IESG review.
- o The range of 60,000 to 99,999 is reserved for experimental YANG modules. This range MUST NOT be used in operational deployments since these SIDs are not globally unique which limit their interoperability. The IANA policy for this range is "Experimental use" [[RFC5226](#)].
- o The range of 100,000 to 999,999 is reserved for standardized YANG modules. The IANA policy for future additions to this sub-registry is "Specification Required" [[RFC5226](#)]. Allocation within this range requires publishing of the associated ".yang" and ".sid" files in the YANG module registry.

Entry Point	Size	IANA policy
0	1,000	IETF review
1,000	59,000	RFC required
60,000	40,000	Experimental use
100,000	900,000	Specification Required

The size of a SID range assigned to a YANG module should be at least 33% above the current number of YANG items. This headroom allows assignment within the same range of new YANG items introduced by subsequent revisions. A larger SID range size may be requested by the authors if this recommendation is considered insufficient. It is important to note that an extra SID range can be allocated to an existing YANG module if the initial range is exhausted.

7.2.2. "RFC SID range assignment" sub-registry

The name of this sub-registry is "RFC SID range assignment". This sub-registry of "IANA SID Mega-Range" allocation [Section 7.2.1](#) corresponds to the SID entry point 1000, size 59000. Each entry in this sub-registry must include:

- o The SID range entry point.
- o The SID range size.
- o The YANG module name.
- o The RFC number.

Initial entries in this registry are as follows:

```

+-----+-----+-----+-----+
| Entry Point | Size | Module name      | RFC number      |
+-----+-----+-----+-----+
| 1000        | 100  | ietf-comi        | [I-D.ietf-core-comi] |
| 1100        | 50   | ietf-yang-types  | [RFC6021]             |
| 1150        | 50   | ietf-inet-types  | [RFC6021]             |
| 1200        | 50   | iana-crypt-hash  | [RFC7317]             |
| 1250        | 50   | ietf-netconf-acm | [RFC6536]             |
| 1300        | 50   | ietf-sid-file    | RFCXXXX         |
| 1500        | 100  | ietf-interfaces  | [RFC7223]             |
| 1600        | 100  | ietf-ip           | [RFC7277]             |
| 1700        | 100  | ietf-system      | [RFC7317]             |
| 1800        | 400  | iana-if-type     | [RFC7224]             |
+-----+-----+-----+-----+

```

// RFC Ed.: replace XXXX with RFC number assigned to this draft.

For allocation, RFC publication of the module is required as per [\[RFC5226\]](#). The YANG module must be registered in the "YANG module Name" registry according to the rules specified in [section 14 of \[RFC6020\]](#).

7.2.3. "Specification SID range assignment" sub-registry

The name of this sub-registry is "Specification SID range assignment". This sub-registry of "IANA SID Mega-Range" allocation [Section 7.2.1](#) corresponds to the SID entry point 100000, size 900000. Each entry in this sub-registry must include:

- o The SID range entry point.

- o The SID range size.
- o The YANG module name.
- o The name of the standard organization
- o The specification identifier or URI

7.3. "YANG module assignment" registry

The name of this registry is "YANG module assignment". This registry is used to track which YANG modules have been assigned and the specific YANG items assignment. Each entry in this registry must include:

- o The YANG module name.
- o The associated ".yang" file(s)
- o The associated ".sid" file

The validity of the ".yang" and ".sid" files added to this registry MUST be verified.

- o The syntax of the registered ".yang" and ".sid" files must be valid.
- o Each YANG item defined by the registered ".yang" file must have a corresponding SID assigned in the ".sid" file.
- o Each SID is assigned to a single YANG item, duplicate assignment is not allowed.
- o The SID range(s) defined in the ".sid" file must be unique, must not conflict with any other SID ranges defined in already registered ".sid" files.
- o The ownership of the SID range(s) should be verified.

The IANA policy for future additions to this registry is "First Come First Served" as described in [[RFC5226](#)].

8. Acknowledgments

The authors would like to thank Andy Bierman, Carsten Bormann, Abhinav Somaraju, Laurent Toutain, Randy Turner and Peter van der Stok for their help during the development of this document and their useful comments during the review process.

9. References

9.1. 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, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC7049] Bormann, C. and P. Hoffman, "Concise Binary Object Representation (CBOR)", [RFC 7049](#), DOI 10.17487/RFC7049, October 2013, <<https://www.rfc-editor.org/info/rfc7049>>.
- [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>>.

9.2. Informative References

- [I-D.ietf-core-comi] Veillette, M., Stok, P., Pelov, A., and A. Bierman, "CoAP Management Interface", [draft-ietf-core-comi-04](#) (work in progress), November 2018.
- [RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", [RFC 5226](#), DOI 10.17487/RFC5226, May 2008, <<https://www.rfc-editor.org/info/rfc5226>>.
- [RFC6020] Bjorklund, M., Ed., "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", [RFC 6020](#), DOI 10.17487/RFC6020, October 2010, <<https://www.rfc-editor.org/info/rfc6020>>.
- [RFC6021] Schoenwaelder, J., Ed., "Common YANG Data Types", [RFC 6021](#), DOI 10.17487/RFC6021, October 2010, <<https://www.rfc-editor.org/info/rfc6021>>.
- [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, <<https://www.rfc-editor.org/info/rfc6241>>.

- [RFC6536] Bierman, A. and M. Bjorklund, "Network Configuration Protocol (NETCONF) Access Control Model", [RFC 6536](#), DOI 10.17487/RFC6536, March 2012, <<https://www.rfc-editor.org/info/rfc6536>>.
- [RFC7223] Bjorklund, M., "A YANG Data Model for Interface Management", [RFC 7223](#), DOI 10.17487/RFC7223, May 2014, <<https://www.rfc-editor.org/info/rfc7223>>.
- [RFC7224] Bjorklund, M., "IANA Interface Type YANG Module", [RFC 7224](#), DOI 10.17487/RFC7224, May 2014, <<https://www.rfc-editor.org/info/rfc7224>>.
- [RFC7277] Bjorklund, M., "A YANG Data Model for IP Management", [RFC 7277](#), DOI 10.17487/RFC7277, June 2014, <<https://www.rfc-editor.org/info/rfc7277>>.
- [RFC7317] Bierman, A. and M. Bjorklund, "A YANG Data Model for System Management", [RFC 7317](#), DOI 10.17487/RFC7317, August 2014, <<https://www.rfc-editor.org/info/rfc7317>>.
- [RFC8040] Bierman, A., Bjorklund, M., and K. Watsen, "RESTCONF Protocol", [RFC 8040](#), DOI 10.17487/RFC8040, January 2017, <<https://www.rfc-editor.org/info/rfc8040>>.

Appendix A. ".sid" file example

The following .sid file (ietf-system@2014-08-06.sid) have been generated using the following yang modules:

- o ietf-system@2014-08-06.yang
- o ietf-yang-types@2013-07-15.yang
- o ietf-inet-types@2013-07-15.yang
- o ietf-netconf-acm@2012-02-22.yang
- o iana-crypt-hash@2014-04-04.yang

```
{
  "assignment-ranges": [
    {
      "entry-point": 1700,
      "size": 100
    }
  ],
  "module-name": "ietf-system",
```



```
"module-revision": "2014-08-06",
"items": [
  {
    "namespace": "module",
    "identifier": "ietf-system",
    "sid": 1700
  },
  {
    "namespace": "identity",
    "identifier": "authentication-method",
    "sid": 1701
  },
  {
    "namespace": "identity",
    "identifier": "local-users",
    "sid": 1702
  },
  {
    "namespace": "identity",
    "identifier": "radius",
    "sid": 1703
  },
  {
    "namespace": "identity",
    "identifier": "radius-authentication-type",
    "sid": 1704
  },
  {
    "namespace": "identity",
    "identifier": "radius-chap",
    "sid": 1705
  },
  {
    "namespace": "identity",
    "identifier": "radius-pap",
    "sid": 1706
  },
  {
    "namespace": "feature",
    "identifier": "authentication",
    "sid": 1707
  },
  {
    "namespace": "feature",
    "identifier": "dns-udp-tcp-port",
    "sid": 1708
  },
  {

```



```
    "namespace": "feature",
    "identifier": "local-users",
    "sid": 1709
  },
  {
    "namespace": "feature",
    "identifier": "ntp",
    "sid": 1710
  },
  {
    "namespace": "feature",
    "identifier": "ntp-udp-port",
    "sid": 1711
  },
  {
    "namespace": "feature",
    "identifier": "radius",
    "sid": 1712
  },
  {
    "namespace": "feature",
    "identifier": "radius-authentication",
    "sid": 1713
  },
  {
    "namespace": "feature",
    "identifier": "timezone-name",
    "sid": 1714
  },
  {
    "namespace": "data",
    "identifier": "/ietf-system:set-current-datetime",
    "sid": 1715
  },
  {
    "namespace": "data",
    "identifier": "/ietf-system:set-current-datetime/
      current-datetime",
    "sid": 1716
  },
  {
    "namespace": "data",
    "identifier": "/ietf-system:system",
    "sid": 1717
  },
  {
    "namespace": "data",
    "identifier": "/ietf-system:system-restart",
```



```
    "sid": 1718
  },
  {
    "namespace": "data",
    "identifier": "/ietf-system:system-shutdown",
    "sid": 1719
  },
  {
    "namespace": "data",
    "identifier": "/ietf-system:system-state",
    "sid": 1720
  },
  {
    "namespace": "data",
    "identifier": "/ietf-system:system-state/clock",
    "sid": 1721
  },
  {
    "namespace": "data",
    "identifier": "/ietf-system:system-state/clock/boot-datetime",
    "sid": 1722
  },
  {
    "namespace": "data",
    "identifier": "/ietf-system:system-state/clock/
      current-datetime",
    "sid": 1723
  },
  {
    "namespace": "data",
    "identifier": "/ietf-system:system-state/platform",
    "sid": 1724
  },
  {
    "namespace": "data",
    "identifier": "/ietf-system:system-state/platform/machine",
    "sid": 1725
  },
  {
    "namespace": "data",
    "identifier": "/ietf-system:system-state/platform/os-name",
    "sid": 1726
  },
  {
    "namespace": "data",
    "identifier": "/ietf-system:system-state/platform/os-release",
    "sid": 1727
  },
},
```



```
{
  "namespace": "data",
  "identifier": "/ietf-system:system-state/platform/os-version",
  "sid": 1728
},
{
  "namespace": "data",
  "identifier": "/ietf-system:system/authentication",
  "sid": 1729
},
{
  "namespace": "data",
  "identifier": "/ietf-system:system/authentication/user",
  "sid": 1730
},
{
  "namespace": "data",
  "identifier": "/ietf-system:system/authentication/
    user-authentication-order",
  "sid": 1731
},
{
  "namespace": "data",
  "identifier": "/ietf-system:system/authentication/user/
    authorized-key",
  "sid": 1732
},
{
  "namespace": "data",
  "identifier": "/ietf-system:system/authentication/user/
    authorized-key/algorithm",
  "sid": 1733
},
{
  "namespace": "data",
  "identifier": "/ietf-system:system/authentication/user/
    authorized-key/key-data",
  "sid": 1734
},
{
  "namespace": "data",
  "identifier": "/ietf-system:system/authentication/user/
    authorized-key/name",
  "sid": 1735
},
{
  "namespace": "data",
  "identifier": "/ietf-system:system/authentication/user/
```



```
        name",
    "sid": 1736
  },
  {
    "namespace": "data",
    "identifier": "/ietf-system:system/authentication/user/
        password",
    "sid": 1737
  },
  {
    "namespace": "data",
    "identifier": "/ietf-system:system/clock",
    "sid": 1738
  },
  {
    "namespace": "data",
    "identifier": "/ietf-system:system/clock/timezone-name",
    "sid": 1739
  },
  {
    "namespace": "data",
    "identifier": "/ietf-system:system/clock/timezone-utc-offset",
    "sid": 1740
  },
  {
    "namespace": "data",
    "identifier": "/ietf-system:system/contact",
    "sid": 1741
  },
  {
    "namespace": "data",
    "identifier": "/ietf-system:system/dns-resolver",
    "sid": 1742
  },
  {
    "namespace": "data",
    "identifier": "/ietf-system:system/dns-resolver/options",
    "sid": 1743
  },
  {
    "namespace": "data",
    "identifier": "/ietf-system:system/dns-resolver/options/
        attempts",
    "sid": 1744
  },
  {
    "namespace": "data",
    "identifier": "/ietf-system:system/dns-resolver/options/
```



```
        timeout",
    "sid": 1745
},
{
    "namespace": "data",
    "identifier": "/ietf-system:system/dns-resolver/search",
    "sid": 1746
},
{
    "namespace": "data",
    "identifier": "/ietf-system:system/dns-resolver/server",
    "sid": 1747
},
{
    "namespace": "data",
    "identifier": "/ietf-system:system/dns-resolver/server/name",
    "sid": 1748
},
{
    "namespace": "data",
    "identifier": "/ietf-system:system/dns-resolver/server/
        udp-and-tcp",
    "sid": 1749
},
{
    "namespace": "data",
    "identifier": "/ietf-system:system/dns-resolver/server/
        udp-and-tcp/address",
    "sid": 1750
},
{
    "namespace": "data",
    "identifier": "/ietf-system:system/dns-resolver/server/
        udp-and-tcp/port",
    "sid": 1751
},
{
    "namespace": "data",
    "identifier": "/ietf-system:system/hostname",
    "sid": 1752
},
{
    "namespace": "data",
    "identifier": "/ietf-system:system/location",
    "sid": 1753
},
{
    "namespace": "data",
```



```
    "identifier": "/ietf-system:system/ntp",
    "sid": 1754
  },
  {
    "namespace": "data",
    "identifier": "/ietf-system:system/ntp/enabled",
    "sid": 1755
  },
  {
    "namespace": "data",
    "identifier": "/ietf-system:system/ntp/server",
    "sid": 1756
  },
  {
    "namespace": "data",
    "identifier": "/ietf-system:system/ntp/server/
      association-type",
    "sid": 1757
  },
  {
    "namespace": "data",
    "identifier": "/ietf-system:system/ntp/server/iburst",
    "sid": 1758
  },
  {
    "namespace": "data",
    "identifier": "/ietf-system:system/ntp/server/name",
    "sid": 1759
  },
  {
    "namespace": "data",
    "identifier": "/ietf-system:system/ntp/server/prefer",
    "sid": 1760
  },
  {
    "namespace": "data",
    "identifier": "/ietf-system:system/ntp/server/udp",
    "sid": 1761
  },
  {
    "namespace": "data",
    "identifier": "/ietf-system:system/ntp/server/udp/address",
    "sid": 1762
  },
  {
    "namespace": "data",
    "identifier": "/ietf-system:system/ntp/server/udp/port",
    "sid": 1763
  }
}
```



```
    },
    {
      "namespace": "data",
      "identifier": "/ietf-system:system/radius",
      "sid": 1764
    },
    {
      "namespace": "data",
      "identifier": "/ietf-system:system/radius/options",
      "sid": 1765
    },
    {
      "namespace": "data",
      "identifier": "/ietf-system:system/radius/options/attempts",
      "sid": 1766
    },
    {
      "namespace": "data",
      "identifier": "/ietf-system:system/radius/options/timeout",
      "sid": 1767
    },
    {
      "namespace": "data",
      "identifier": "/ietf-system:system/radius/server",
      "sid": 1768
    },
    {
      "namespace": "data",
      "identifier": "/ietf-system:system/radius/server/
        authentication-type",
      "sid": 1769
    },
    {
      "namespace": "data",
      "identifier": "/ietf-system:system/radius/server/name",
      "sid": 1770
    },
    {
      "namespace": "data",
      "identifier": "/ietf-system:system/radius/server/udp",
      "sid": 1771
    },
    {
      "namespace": "data",
      "identifier": "/ietf-system:system/radius/server/udp/
        address",
      "sid": 1772
    },
  },
```



```
{
  "namespace": "data",
  "identifier": "/ietf-system:system/radius/server/udp/
    authentication-port",
  "sid": 1773
},
{
  "namespace": "data",
  "identifier": "/ietf-system:system/radius/server/udp/
    shared-secret",
  "sid": 1774
}
]
```

Authors' Addresses

Michel Veillette (editor)
Trilliant Networks Inc.
610 Rue du Luxembourg
Granby, Quebec J2J 2V2
Canada

Phone: +14503750556
Email: michel.veillette@trilliant.com

Alexander Pelov (editor)
Acklio
1137A avenue des Champs Blancs
Cesson-Sevigne, Bretagne 35510
France

Email: a@ackl.io

Ivaylo Petrov (editor)
Acklio
1137A avenue des Champs Blancs
Cesson-Sevigne, Bretagne 35510
France

Email: ivaylo@ackl.io

