

CoRE
Internet-Draft
Updates: RFC9254 (if approved)
Intended status: Standards Track
Expires: 3 September 2026

V. Vilimek
CZ.NIC
2 March 2026

Encoding rules of YANG instance-identifier in the Concise Binary Object
Representation (CBOR)
draft-vilimek-yang-cbor-inst-id-01

Abstract

The YANG-CBOR document [RFC9254] defines rules for representing YANG-modeled data [RFC7950] in CBOR [RFC8949]. The YANG-CBOR defines rules for all built-in types, such as int64, leafref, and instance-identifier data type. However it fails to address some cases of the instance-identifier, specifically those pointing to keyless list entries or leaf-list entries. This documents updates [RFC9254] to make the rules complete.

About This Document

This note is to be removed before publishing as an RFC.

Status information for this document may be found at
<https://datatracker.ietf.org/doc/draft-vilimek-yang-cbor-inst-id/>.

Discussion of this document takes place on the Constrained RESTful Environments (CoRE) Working Group mailing list (<mailto:core@ietf.org>), which is archived at <https://mailarchive.ietf.org/arch/browse/core/>. Subscribe at <https://www.ietf.org/mailman/listinfo/core/>.

Source for this draft and an issue tracker can be found at
<https://github.com/vvilimek/draft-vilimek-yang-cbor-inst-id>.

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 3 September 2026.

Copyright Notice

Copyright (c) 2026 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 Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

Table of Contents

1. Introduction	3
2. Terminology and Notation	3
3. Representing SIDs as YANG instance-identifier in CBOR	4
3.1. Motivation example	4
3.2. Rules	5
3.3. Examples	5
3.3.1. Simple leaf-list instance-identifier	6
3.3.2. Leaf-list inside list	6
3.3.3. List instance	7
3.3.4. Instance-identifier chaining	7
3.3.5. Complex example of mixed keyed and keyless lists	8
4. Content-Types	9
5. Security Considerations	10
6. IANA Considerations	10
7. References	10
7.1. Normative References	10
7.2. Informative References	11
Acknowledgments	11
Author's Address	11

1. Introduction

The [RFC9254]: Encoding Rules of Data Modeled with YANG in the Concise Binary Object Representation (CBOR) defines rules for representing of YANG-modeled data [RFC7950] in CBOR [RFC8949]. The instance-identifier YANG data type is well-defined for instances that reference containers, lists with one or more keys, and leafs. The document fails to address cases when the instance-identifier instance points to entries in a list with no keys or a leaf-list. The YANG-CBOR refines encoding rules of two kinds. One variant, named as using names, is compatible with YANG-JSON [RFC7951] encoding. The second variant uses SIDs to compress the information needed to encode the instance-identifier. Only the SID-based encoding rules are incomplete. This documents aims to define the missing rules and to clarify the terminology used.

2. Terminology and Notation

The key words "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] [RFC8174] when, and only when, they appear in all capitals, as shown here.

The following terms are defined in [RFC7950]:

- * list
- * leaf-list
- * leaf
- * container
- * instance-identifier

The following term is defined in [RFC8949]:

- * data item

The following terms are defined in [RFC9254]:

- * YANG Scheme Item iDentifier (YANG SID, or simply SID)
- * delta (of YANG SIDs)
- * absolute SID

The following terms are defined in [RFC9595]:

* item

Keyless list: Is config false YANG list without any keys.

Keyed list: Is YANG list that is not a keyless list. It is either a config true list or a config false list with at least one key.

Note that a config true list must have at least one key.

3. Representing SIDs as YANG instance-identifier in CBOR

3.1. Motivation example

Definition example adapted from [RFC7950]:

```
container system {  
  ...  
  leaf reporting-entity {  
    type instance-identifier;  
  }  
}
```

The following example shows the encoding of the 'reporting-entity' value referencing "neighbor-sysid" (which is assumed to have SID 68000) of the second list entry of the keyless "/isis:adjacencies/adjacency" list. The example is adapted from [RFC9130] and therefore uses the isis namespace:

```
// in module isis  
container adjacencies {  
  config false;  
  list adjacency {  
    leaf neighbor-sysid { // SID 68000  
      type string;  
    }  
    leaf more-data {  
      type binary;  
    }  
  }  
}
```

We want to represent reporting-entity so that the value is "/isis:adjacencies/adjacency[.=2]/neighbor-sysid" instance-identifier, but we want to use SIDs Because the YANG list does not specify any keys, we will use index-based identification. Because the YANG do not specify any keys, we will resort to index based identification.

CBOR diagnostic notation: [68000, 2]

CBOR encoding:

```
82          # array(2)
  1A 000109A0 # 68000
  02          # 2
```

Where 68000 is absolute SID and 2 is 1-based index of the list entry.

3.2. Rules

The definitions of Section 6.13.1 of [RFC9254] require that the list entries are encoded as CBOR array with SID as first element, followed by keys in order same as walk from schema root to the targeted child node. This formulation only applies to keyed lists.

For keyless lists, the entry identification MUST be encoded using a CBOR unsigned integer data item (major type 0). The index MUST be 1-based to maintain the same indexing base as RESTCONF [RFC8040] and NETCONF [RFC6241].

If the target data node is a leaf-list entry, the last element of the CBOR array MUST be encoded according to the encoding rules for the given leaf-list's type.

The YANG 1.1 [RFC7950] allows a leaf-list of state data (config false) to have duplicates. In this case, it is not defined which element the instance-identifier identifies.

Due to restriction set forth in Section 9.13 of [RFC7950] it is not possible to point the instance-identifier to a list or a leaf-list schema node. Only instance data (i.e. nodes in the data tree) can be targeted.

TODO: If these encoding rules are accepted by the working group consensus, changes to the draft I-D.core-comi (CORECONF) will be necessary. This will allow us to identify keyless list instances, which is impossible in RESTCONF.

3.3. Examples

Unless stated otherwise, all examples are considered to live inside the example module namespace. An equivalent representation using the Names encoding may help readers who are already familiar with YANG JSON encoding [RFC7951], or similar XML encoding defined in YANG 1.1 [RFC7950].

3.3.1. Simple leaf-list instance-identifier

The following example shows the encoding of the 'reporting-entity' value referencing leaf-list instance `"/auth/foreign-user"` (which is assumed to have SID 60000) entry `"alice"`. The name-based representation is `"/example:auth/foreign-user[.="alice"]"`.

```
container auth {  
  leaf-list foreign-user { // SID 60000  
    type string;  
  }  
}
```

CBOR diagnostic notation: [60000, "alice"]

CBOR encoding:

```
82          # array(2)  
 19 F6F6    # unsigned(60000)  
 65         # text(5)  
 616c696365 # "alice"
```

3.3.2. Leaf-list inside list

The following example shows the encoding of the 'reporting-entity' value referencing leaf-list instance `"/user-group/user"` (which is assumed to have SID 61000) entry `"eve"` for group-name `"restricted"`. The name-based representation is `"/example:user-group[group-name="restricted"]/user[.="eve"]"`.

```
list user-group {  
  config true;  
  key "group-name"  
  
  leaf group-name {  
    type string;  
  }  
  
  leaf-list user { // SID 61000  
    type string;  
  }  
}
```

CBOR diagnostic notation: [61000, "restricted", "eve"]

CBOR encoding:

```

83          # array(3)
  19 EE48   # 61000
  6A        # text(10)
    72657374726963746564 # "resricted"
  63        # text(3)
    657665   # "eve"

```

3.3.3. List instance

We reuse the YANG snippet from Section 3.3.2. The following example shows the encoding of 'reporting-entity' value referencing list instance `"/user-group"` for group name `"restricted"`. The name-based representation is `"/example:user-group[group-name="restricted"]"`.

CBOR diagnostic notation: [61000, "restricted"]

CBOR encoding:

```

83          # array(3)
  19 EE48   # 61000
  6A        # text(10)
    72657374726963746564 # "resricted"

```

3.3.4. Instance-identifier chaining

The following example shows the encoding of 'reporting-entity' value referencing leaf-list instance `"/working-group/chair"` entry. This entry references `"/auth/foreign-user"` leaf-list entry `"Carsten Bormann"`. The `"/working-group/chair"` is assumed to have SID 62000. The name-based representation is `"/example:working-group[name="core"]/chair[.="example:auth/foreign-user[.="John Smith"]]"`.

```

list working-group {
  leaf name {
    type string;
  }

  leaf-list chair { // SID 62000
    type instance-identifier;
  }
}

```

CBOR diagnostic notation: [62000, "core", [60000, "Carsten Bormann"]]

CBOR encoding:

```

83                                     # array(3)
  19 F230                             # 62000
  64                                     # text(4)
    636F7265                           # "core"
  82                                     # array(2)
    19 F6F6                             # 60000
    6F                                     # text(15)
      4361727374656E20426F726D616E6E   # "Carsten Bormann"

```

TODO longer chains of leaf-list instance-identifier lead to high nesting of the CBOR array data items. Should a cap for the constrained nodes be put to simplify the implementations? I think cap around 8 should be sufficient for most deployments. I think that using leaf-list instance-identifier chaining is not a good practise.

3.3.5. Complex example of mixed keyed and keyless lists

The following example shows the encoding of 'reporting-entity' value referencing leaf 'token-data' of device with 'id' "id01", first 'security' list entry for user's 'bob' second 'access-token' list entry. The leaf 'token-data' is assumed to have SID 61500. The name-based representation is

```
"/example:device[id="id01"]/security[.=1]/user[user="bob"]/access-token[.=2]/token-data".
```

```

list device {
  key "id";

  leaf id {
    type string;
  }

  list security {
    config false;

    list user {
      key "name";
      leaf name;

      list access-token {
        leaf type {
          type identityref { base token; }
        }
        leaf token-data { // SID 61500
          type binary;
        }
      }
    }
  }
}

```

identity token;

CBOR diagnostic notation: [61500, "id01", 1, "bob", 2]

CBOR encoding:

```

84          # array(4)
 19 F03C    # 61500
 64          # text(4)
   69643031 # "id01"
 01          # 1
 63          # text(3)
   626F62   # "bob"
 02          # 2

```

4. Content-Types

TODO Is it possible to reuse the Content-types define in the [RFC9254]? It would be wasteful to assign new MIME content-type to basically the same format.

5. Security Considerations

The security considerations of [RFC8949], [RFC7950], [RFC9254] and [RFC9595] apply.

The implementations should be aware of possibly very large recursive nesting that occurs when instance-identifiers are chained.

TODO Security

6. IANA Considerations

This document has no IANA actions.

TODO Is it possible to keep the same IANA allocations of th [RFC9254]? This draft wants to be more of a bugfix document than new encoding scheme.

7. References

7.1. 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/rfc/rfc2119>>.
- [RFC7950] Bjorklund, M., Ed., "The YANG 1.1 Data Modeling Language", RFC 7950, DOI 10.17487/RFC7950, August 2016, <<https://www.rfc-editor.org/rfc/rfc7950>>.
- [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/rfc/rfc8174>>.
- [RFC8949] Bormann, C. and P. Hoffman, "Concise Binary Object Representation (CBOR)", STD 94, RFC 8949, DOI 10.17487/RFC8949, December 2020, <<https://www.rfc-editor.org/rfc/rfc8949>>.
- [RFC9254] Veillette, M., Ed., Petrov, I., Ed., Pelov, A., Bormann, C., and M. Richardson, "Encoding of Data Modeled with YANG in the Concise Binary Object Representation (CBOR)", RFC 9254, DOI 10.17487/RFC9254, July 2022, <<https://www.rfc-editor.org/rfc/rfc9254>>.

- [RFC9595] Veillette, M., Ed., Pelov, A., Ed., Petrov, I., Ed., Bormann, C., and M. Richardson, "YANG Schema Item Identifier (YANG SID)", RFC 9595, DOI 10.17487/RFC9595, July 2024, <<https://www.rfc-editor.org/rfc/rfc9595>>.

7.2. Informative References

- [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/rfc/rfc6241>>.
- [RFC7951] Lhotka, L., "JSON Encoding of Data Modeled with YANG", RFC 7951, DOI 10.17487/RFC7951, August 2016, <<https://www.rfc-editor.org/rfc/rfc7951>>.
- [RFC8040] Bierman, A., Bjorklund, M., and K. Watsen, "RESTCONF Protocol", RFC 8040, DOI 10.17487/RFC8040, January 2017, <<https://www.rfc-editor.org/rfc/rfc8040>>.
- [RFC9130] Litkowski, S., Ed., Yeung, D., Lindem, A., Zhang, J., and L. Lhotka, "YANG Data Model for the IS-IS Protocol", RFC 9130, DOI 10.17487/RFC9130, October 2022, <<https://www.rfc-editor.org/rfc/rfc9130>>.

Acknowledgments

TODO acknowledge. thank Andy Bierman for his friendly responses on mailing list.

Author's Address

Vojtěch Vilímek
CZ.NIC
Milesovska 1136/5
13000 Praha
Czech Republic
Email: vojtech.vilimek@nic.cz