

WG Working Group  
Internet-Draft  
Updates: RFC9254 (if approved)  
Intended status: Standards Track  
Expires: 6 May 2026

V. Vilimek  
CZ.NIC  
2 November 2025

Encoding rules of YANG 'instance-identifier' in the Concise Binary  
Object Representation (CBOR)  
draft-vilimek-yang-cbor-inst-id-00

## Abstract

Encoding rules of YANG-CBOR [RFC9254] are incomplete for 'instance-identifier' YANG data type. This document defines missing encoding rules for this data type.

## 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 6 May 2026.

## Copyright Notice

Copyright (c) 2025 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 . . . . .	2
2. Terminology and Notation . . . . .	2
3. Representing YANG 'instance-identifier' Type in CBOR . . . . .	4
3.1. SIDs as 'instance-identifier' . . . . .	4
3.1.1. Examples . . . . .	5
4. Content-Types . . . . .	9
5. Security Considerations . . . . .	9
6. IANA Considerations . . . . .	10
7. References . . . . .	10
7.1. Normative References . . . . .	10
7.2. Informative References . . . . .	10
Acknowledgments . . . . .	11
Author's Address . . . . .	11

## 1. Introduction

The RFC 9254 Encoding Rules of Data Modeled with YANG in the Concise Binary Object Representation (CBOR) does not define encoding rules for 'instance-identifier' pointing to list without keys entry instances and instances of leaf-list entries. The goal of this document is to define the missing rules and make clarifications in the used terminology.

## 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 defiend 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]:

- \* delta (of YANG SIDs)
- \* absolute SID

The following terms are defined in [RFC9595]:

- \* item
- \* YANG Schema Item iDentifier (or "YANG SID" or simply "SID")

Note that the [RFC9254] also define term YANG Schema Item iDentifier but the definition describe the same term.

TODO: use the "The following terms are used within this document:" header?

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 config false list with at least one key.

Single instance node: Is a instance node with at most one possible

instantiation. Instantiations of top-level containers are single instance nodes, instantiations of leafs of toplevel containers are single instance nodes. Container and leaf instantiations of single instance node are also single instance nodes. No list or leaf-list entries are single instance nodes, even if they have max-elements equal to one. If instance is a child of list entry it is not a single instance node. Note that this term is defined so that set of instance nodes that are uniquely identified by only YANG Schema Item identifier and set of single instance nodes are the same set.

### 3. Representing YANG 'instance-identifier' Type in CBOR

#### 3.1. SIDs as 'instance-identifier'

The definitions of Section 6.13.1 of [RFC9254] applies with following exceptions:

The encoding rules for list apply only for keyed lists.

In the case of a representation node that is an entry of a keyless list, a SID is combined with the list entry index is used to identify each instance within the keyless list. The index MUST be encoded using CBOR unsigned integer data item (major type 0). The index MUST be 1-base to keep same indexing base as RESTCONF [RFC8040] and NETCONF [RFC6241].

Instance-identifier of an instance that is not single instance node MUST be encoded using a CBOR array item (major type 4) containing the following CBOR data items:

- \* The first element MUST be encoded as a CBOR unsigned integer data item (major type 0) and set to the targeted schema node SID. No delta mechanism for SID is used.
- \* The next elements MUST contain the value of each key required to identify the instance of the targeted schema node. These keys MUST be ordered as defined in the 'key' YANG statement for keyed list. The keys are encoded according the rules defined in [RFC9254] and this document. If the list is keyless list the key MUST be encoded using the CBOR unsigned integer data item (major type 0) as specified in this document. The order of the keys and indices MUST be same as walk from top-level node down to targeted schema node.
- \* If the instance is leaf-list entry, the last element MUST be encoded according to encoding rules defined in [RFC9254] and this document.

This means that instance-identifier identifying a leaf-list instance with single instance node parent will result in a CBOR array with two elements, the SID as CBOR unsigned integer and leaf-list value representation.

TODO: is this a good solution?

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

### 3.1.1.1. Examples

Definition example adapted from [RFC7950]:

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

YANG model code snippet used for second and third example:

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

All examples are considered to live inside the example module namespace if not stated otherwise. Equivalent representation using the Names encoding may help readers already familiar with YANG JSON encoding [RFC7951], or similar XML encoding defined in YANG 1.1 [RFC7950].

\_First example:\_

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

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

CBOR diagnostic notation: [ 68000, 2 ]

CBOR encoding: 82 # array(2) 1A 000109A0 # 68000 02 # 2

Equivalent instance-identifier encoded using the Names:  
"/isis:adjacencies/adjacency[.=2]/neighbor-sysid"

Second example:

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".

CBOR diagnostic notation: [ 60000, "alice" ]

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

Equivalent instance-identifier encoded using the Names:  
"/example:auth/foreign-user[.="alice"]"

Third example:

The following example show the encoding of the 'reporting-entity' value referencing leaf-list instance "/auth/foreign-user" (SID 60000).

CBOR diagnostic notation: 60000

CBOR encoding: 19 F6F6

Equivalent instance-identifier encoded using the Names:  
"/example:auth/foreign-user"

Fourth example:

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"`.

```
list user-group {
  config true;
  key "group-name"

  leaf group-name {
    type string;
  }

  leaf-list user {
    type string;
  }
}
```

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

CBOR encoding: 83 # array(3) 19 EE48 # 61000 6A # text(10)  
72657374726963746564 # "restricted" 63 # text(3) 657665 # "eve"

Equivalent instance-identifier encoded using the Names:  
`"/example:user-group[group-name="restricted"]/user[.="eve"]"`

\_Fifth example:\_

The following example shows the encoding of 'reporting-entity' value referencing leaf-list instance `"/user-group/user"` for group name `"restricted"`.

CBOR diagnostic notation: [ 61000, "restricted" ]

CBOR encoding: 83 # array(3) 19 EE48 # 61000 6A # text(10)  
72657374726963746564 # "restricted"

Equivalent instance-identifier encoded using the Names:  
`"/example:user-group[group-name="restricted"]"`

Note that this encoding is same as if the node `user` was a leaf.

\_Sixth example:\_

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 `"John Smith"`. The `"/working-group/chair"` is assumed to have SID 62000.

```
list working-group {  
  leaf name {  
    type string;  
  }  
  
  leaf-list chair {  
    type instance-identifier;  
  }  
}
```

CBOR diagnostic notation: [ 62000, "core", [ 60000, "John Smith" ] ]

CBOR encoding: 83 # array(3) 19 F230 # 62000 64 # text(4) 636F7265 #  
"core" 82 # array(2) 19 F6F6 # 60000 6A # text(10)  
4a6f686e20536d697468 # "John Smith"

Equivalent instance-identifier encoded using the Names:

"/example:working-group[name="core"]/chair=[.="/example:auth/foreign-  
user[.="John Smith"]"]

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

\_Seventh example:\_

The following exampke 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.



```
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 {
          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

Equivalent instance-identifier encoded using the Names:  
"/example:device[id="id01"]/security[.=1]/user[user="bob"]/access-  
token[.=2]/token-data"

#### 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  
basically the same format.

#### 5. Security Considerations

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

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](mailto:vojtech.vilimek@nic.cz)