Workgroup:
Concise Binary Object Representation
Maintenance and Extensions
Internet-Draft:
draft-bormann-cbor-yang-standin-00
Published: 21 February 2024
Intended Status: Standards Track
Expires: 24 August 2024
Authors: C. Bormann
Universität Bremen TZI
M. Matejka
CZ.NIC

Stand-in Tags for YANG-CBOR

Abstract

YANG (RFC 7950) is a data modeling language used to model configuration data, state data, parameters and results of Remote Procedure Call (RPC) operations or actions, and notifications.

YANG-CBOR (RFC 9254) defines encoding rules for YANG in the Concise Binary Object Representation (CBOR) (RFC 8949). While the overall structure of YANG-CBOR is encoded in an efficient, binary format, YANG itself has its roots in XML and therefore traditionally encodes some information such as date/times and IP addresses/prefixes in a verbose text form.

This document defines how to use existing CBOR tags for this kind of information in YANG-CBOR as a "stand-in" for the text-based information that would be found in the original form of YANG-CBOR.

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-bormann-cbor-yang-standin/.

Discussion of this document takes place on the CBOR (Concise Binary Object Representation Maintenance and Extensions) Working Group mailing list (mailto:cbor@ietf.org), which is archived at https://mailarchive.ietf.org/arch/browse/cbor/. Subscribe at https://www.ietf.org/mailman/listinfo/cbor/.

Source for this draft and an issue tracker can be found at https://github.com/cabo/yang-standin.

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 24 August 2024.

Copyright Notice

Table of Contents

1. Introduction

(see abstract)

2. Conventions and Definitions

The terminology of [RFC9254] applies.

Legacy representation: The (often text-based) representation for a YANG data item as used in YANG-XML, YANG-JSON, and (unchanged) YANG-CBOR.

Stand-in tag: A CBOR tag that can supply the information that is equivalent to a legacy representation in a more efficient format (e.g., using binary data).

Encoder: The party which generates (sends) CBOR data described by YANG.

Intermediate Encoder: An encoder which isn't the original author of the data, converting it from legacy representation.

Decoder: The party which receives and parses CBOR data described by YANG.

Intermediate Decoder: A decoder which isn't the final recipient of the data, converting it to legacy representation.

Data Transfer: A series of actions, generally beginning by data origination, encoding, continuing by optional intermediate transcoding, sending and receiving, and finally decoding and consuming.

Round Trip: Part of a data transfer between an encoder generating CBOR data with stand-in tags and a decoder parsing the data.

Legacy Round Trip: A Round Trip where the encoder is an intermediate encoder or the decoder is an intermediate decoder and any of these converts from or to the legacy representation.

Unambiguous Round Trip: A Legacy Round Trip that provides exactly the same legacy representation (not just semantically equivalent). The stand-in tag is also said to "unambiguously stand in" for the legacy representation.

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.

3. Stand-In Tags

This document defines two sets of stand-in tags. Where information starts out in a legacy representation, these tags are only used when an Unambiguous Round Trip can be achieved.

3.1. ietf-yang-types: Tag 1 (Date/Time) and Tag 100 (Date)

Section 3 of [I-D.schoenw-netmod-rfc6991-bis] defines the following types in ietf-yang-types:

YANG type	base type	specification	stand-in
date-and-time	string	[RFC6021]	tag 1
date-with-zone-offset	string	[I-D.schoenw-netmod-rfc6991-bis]	(none)
date-no-zone	string	[I-D.schoenw-netmod-rfc6991-bis]	tag 100

Table 1: Legacy representations in ietf-yang-types

Tag 1 (Section 3.4.2 of RFC 8949 [STD94]) can unambiguously stand in for all date-and-time values that:

* do not specify a time zone (note that [I-D.schoenw-netmod-rfc6991-bis] uses the legacy "-00:00" format for time-zone-free date-times)

* are not an inserted leap second (23:59:60 or 23:59:61)

* do not have trailing zeroes in the fractional part of the seconds.

* do not have fractional parts of the seconds with a precision that cannot be represented in floating-point tag content in a tag 1.

All other date-and-time values stay in legacy representation.

Tag 1 uses an integer tag content for all date-and-time values without fractional seconds and a floating-point tag content for values that have fractional seconds given.

Tag 100 [RFC8943] can unambiguously stand in for all date-no-zone values.

3.2. ietf-inet-types: Tags 54 and 52 (IP addresses and prefixes)

Section 4 of [I-D.schoenw-netmod-rfc6991-bis] defines in ietf-inet-types:

YANG type	base type	specification	stand-in
ip-address	union	[RFC6021]	(see union)
ipv6-address	string	[RFC6021]	tag 54
ipv4-address	string	[RFC6021]	tag 52
ip-address-no-zone	union	RFC 6991	(see union)
ipv6-address-no-zone	ipv6-address	RFC 6991	tag 54
ipv4-address-no-zone	ipv4-address	RFC 6991	tag 52
ip-address-link-local	union	[I-D.schoenw-netmod-rfc6991-bis]	(see union)
ipv6-address-link-local	ipv6-address	[I-D.schoenw-netmod-rfc6991-bis]	tag 54
ipv4-address-link-local	ipv4-address	[I-D.schoenw-netmod-rfc6991-bis]	tag 52
ip-prefix	union	[RFC6021]	(see union)
ipv6-prefix	string	[RFC6021]	tag 54
ipv4-prefix	string	[RFC6021]	tag 52
ip-address-and-prefix	union	[I-D.schoenw-netmod-rfc6991-bis]	(see union)
ipv6-address-and-prefix	string	[I-D.schoenw-netmod-rfc6991-bis]	tag 54
ipv4-address-and-prefix	string	[I-D.schoenw-netmod-rfc6991-bis]	tag 52

Table 2: Legacy representations in ietf-yang-types

An intermediate encoder **MAY** normalize IPv6 addresses and prefixes that do not comply with [RFC5952] but can be converted into the stand-in representation. For example, IPv6 address written as 2001:db8:: is the same as 2001:0db8::0:0 and both would be converted to 54(h'20010db800000000000000000000000'), anyway only the first one complies with [RFC5952]. The encoder **MAY** refuse to convert the latter one.

If the schema specifies ip-prefix, an intermediate encoder **MAY** normalize prefixes with non-zero bits after the prefix end. For example, if the legacy representation of ipv6-prefix is 2001:db8:1::/40, the encoder may either refuse it as malformed or convert it to 2001:db8::/40 and represent as 54([40, h'20010db8']).

The encoder implementation should be clear about which normalizations are employed and how.

Adapted examples from [RFC9164]:

Stand-in representation of IPv6 address
2001:db8:1234:deed:beef:cafe:face:feed is
54(h'20010db81234deedbeefcafefacefeed').

CBOR encoding of stand-in (19 bytes):

```
cbor-pretty D8 36 # tag(54) 50 # bytes(16)
20010DB81234DEEDBEEFCAFEFACEFEED
```

CBOR encoding of legacy representation (40 bytes):

```
cbor-pretty 78 26 # text(38)
323030313A6462383A313233343A646565643A626565663A636166653A666163653A6
6656564
```

Stand-in representation of IPv6 prefix 2001:db8:1234::/48 is 54([48,
h'20010db81234']).

CBOR encoding of stand-in (12 bytes):

```
cbor-pretty D8 36 # tag(54) 82 # array(2) 18 30 # unsigned(48) 46 #
bytes(6) 20010DB81234 # " \u0001\r\xB8\u00124"
```

CBOR encoding of legacy representation (19 bytes):

```
cbor-pretty 72 # text(18) 323030313A6462383A313233343A3A2F3438 #
"2001:db8:1234::/48"
```

Stand-in representation of IPv6 link-local address
fe80::0202:02ff:ffff:fe03:0303/64%eth0 is
54([h'fe8000000000020202ffffffe030303', 64, 'eth0']).

CBOR encoding of stand-in (27 bytes):

```
cbor-pretty D8 36 # tag(54) 83 # array(3) 50 # bytes(16)
FE8000000000020202FFFFFFFE030303 18 40 # unsigned(64) 44 # bytes(4)
65746830 # "eth0"
```

CBOR encoding of legacy representation (40 bytes):

```
cbor-pretty 78 26 # text(38)
666538303A3A303230323A303266663A666666663A666530333A303330332F3634256
5746830
```

TO DO: adapt more examples from [RFC9164]

TO DO: Check how the unions in [RFC6021] and
[I-D.schoenw-netmod-rfc6991-bis] interact with this. E.g., the union
ip-address needs to be parsed to decide between tag 54 and tag 52.

3.3. Union handling

When the schema specifies a union data type for a node, there are additional requirements on the encoder and decoder.

An encoder which is fully aware of data semantics **MUST** use the appropriate data type, even though it isn't formally specified by the schema.

If an intermediate encoder doesn't fully understand the data semantics, it needs to find out which type the data actually is to choose the right stand-in. If more types are possible, it **MAY** choose any of these which allow for an Unambiguous Round Trip, otherwise it **SHOULD** keep the legacy representation.

If a decoder receives data for a union-typed node, it **MUST** accept any data type of the union, even though it may violate additional constraints outside the schema.

4. Using Stand-In Tags

4.1. Defining Stand-In Usage in Schema

TO DO: formally define the YANG extension

4.2. Original stand-ins

The simplest situation is when no intermediate encoders and decoders are involved in the data transfer, therefore the round trip is not legacy. In this case, no conversions are involved and data is validated using the schema extension from the previous section.

4.3. Legacy Round Trip

Producing a stand-in **MUST** be triggered by schema usage. Intermediate encoders **MUST NOT** encode stand-ins when no schema is available.

It's generally not recommended to do a legacy round trip where both the encoder and decoder are converting from and to the legacy representation.

5. Negotiation

Introducing stand-in tags in YANG-CBOR requires some form of consent between the producer and the consumer of YANG-CBOR information:

* A producer that creates YANG-CBOR containing stand-in tags needs to know whether the consumer supports stand-in tags, and, possibly, which specific stand-in tags it supports. We speak about the *capability* of a consumer to consume stand-in tags. A producer

MUST NOT employ stand-in tags unless it knows about the capabilities of the consumer. A consumer **SHOULD** indicate its capabilities for consuming stand-in tags.

*A consumer may not want to implement certain legacy text-based representations where more efficient (and easy to implement) stand-in tags are available. This places a *requirement* on the producer (which needs to have the *capability* to produce YANG-CBOR where those stand-in tags are used, in place of legacy representations). A producer **MUST NOT** employ legacy representations where stand-in tags are *required* by the consumer. A consumer that has requirements for only receiving stand-in tags in place of legacy representations, **MUST** indicate this to the producer.

ISSUE: Where do we put those two aspects of negotiation?

*NETCONF negotiation

*yang-library

*media-type parameters

*?

6. Security Considerations

TODO Security

7. IANA Considerations

7.1. stand-in tags?

ISSUE: Do we want to have a separate registry for stand-in tags?

They already are CBOR tags and thus in the in the registry, but might get lost in the bulk of that (and are only identified as YANG-CBOR stand-in Tags in the specification).

7.2. media-type parameters

ISSUE: Should the use of stand-in tags be mentioned in the various YANG-CBOR-based media types (as a media type parameter)? Compare how application/yang-data+cbor can use id=name/id=sid to indicate another encoding decision.

8. Normative References

[I-D.schoenw-netmod-rfc6991-bis]

Schönwälder, J., "Common YANG Data Types", Work in Progress, Internet-Draft, draft-schoenw-netmod-rfc6991-bis-01, 11 March 2019, <https://datatracker.ietf.org/doc/html/draft-schoenw-netmod-rfc6991-bis-01>.

[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>.

[RFC5952] Kawamura, S. and M. Kawashima, "A Recommendation for IPv6 Address Text Representation", RFC 5952, DOI 10.17487/RFC5952, August 2010, <https://www.rfc-editor.org/rfc/rfc5952>.

[RFC6021] Schoenwaelder, J., Ed., "Common YANG Data Types", RFC 6021, DOI 10.17487/RFC6021, October 2010, <https://www.rfc-editor.org/rfc/rfc6021>.

[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>.

[RFC8943] Jones, M., Nadalin, A., and J. Richter, "Concise Binary Object Representation (CBOR) Tags for Date", RFC 8943, DOI 10.17487/RFC8943, November 2020, <https://www.rfc-editor.org/rfc/rfc8943>.

[RFC9164] Richardson, M. and C. Bormann, "Concise Binary Object Representation (CBOR) Tags for IPv4 and IPv6 Addresses and Prefixes", RFC 9164, DOI 10.17487/RFC9164, December 2021, <https://www.rfc-editor.org/rfc/rfc9164>.

[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>.

[STD94] Bormann, C. and P. Hoffman, "Concise Binary Object Representation (CBOR)", STD 94, RFC 8949, December 2020. <https://www.rfc-editor.org/info/std94>

Acknowledgments

TODO acknowledge.

Authors' Addresses

Carsten Bormann

Universität Bremen TZI
Postfach 330440
D-28359 Bremen
Germany

Phone: +49-421-218-63921
Email: cabo@tzi.org

Maria Matejka
CZ.NIC
Milesovska 1136/5
13000 Praha
Czechia

Email: maria.matejka@nic.cz, mq@jmq.cz