

Concise Binary Object Representation (CBOR) Tag for CBOR Templates
draft-bormann-lpwan-cbor-template-02

Abstract

The Concise Binary Object Representation (CBOR, [RFC 7049](#)) is a data format whose design goals include the possibility of extremely small code size, fairly small message size, and extensibility without the need for version negotiation.

The present document makes use of this extensibility to define a CBOR tag for a variable within a CBOR data item, which then could be filled in by a separate process (e.g., from another CBOR data item).

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 July 21, 2018.

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.	Variable	2
2.1.	Example	3
3.	CDDL typename	3
4.	IANA Considerations	4
5.	Security Considerations	4
6.	References	5
6.1.	Normative References	5
6.2.	Informative References	5
	Contributors	5
	Acknowledgements	5
	Author's Address	5

[1.](#) Introduction

The Concise Binary Object Representation (CBOR, [[RFC7049](#)]) provides for the interchange of structured data without a requirement for a pre-agreed schema. [RFC 7049](#) defines a basic set of data types, as well as a tagging mechanism that enables extending the set of data types supported via an IANA registry.

With the work on static compression for CoAP [[RFC7252](#)] and CoAP application protocols [[I-D.ietf-core-comi](#)] on LPWANS [[I-D.ietf-lpwan-coap-static-context-hc](#)], there appears to be a need for defining CBOR data items that have within them some open positions in them that can later be filled in from a separate source (such as another CBOR data item). The anchor points for this substitution are called "variables" in this specification.

This document defines a CBOR tag for a variable in a CBOR data item. It is intended as the reference document for the IANA registration of the tag defined.

[2.](#) Variable

A variable is a CBOR data item that typically is included as a part of a larger data item (the "CBOR template"). In a process that is outside the scope of this specification, the variable is then substituted by an actual value in order to yield an instance from the template.

A variable is identified by the data item within the tag, the "variable identifier". Typically, variables are numbered by integers. Some applications may also benefit from the use of strings as identifiers. The specification of the tag does not make a restriction on the type of the identifier; however note that very complex variable identifiers may benefit from canonicalization to enable their comparison, cf. [section 3.9 of \[RFC7049\]](#).

We term a CBOR data item that contains one or more variables as a "CBOR template"; generally processes that accept CBOR templates with variables will also accept CBOR data items without variables, so we accept this as a degenerate case for "CBOR template". Note that a template may use the same variable (i.e., a variable with the same identifier) in multiple positions, leading to multiple substitutions of the same value.

[2.1.](#) Example

An example for a CBOR template in diagnostic notation:

```
{ "name": "Carsten Bormann",  
  "place": 42(0) }
```

When this template undergoes substitution, with the variable 0 set to the value "Bremen", this would result in the data item:

```
{ "name": "Carsten Bormann",  
  "place": "Bremen" }
```

[3.](#) CDDL typename

CDDL [[I-D.greevenbosch-appsawg-cbor-cddl](#)] definitions will typically describe the structure of a data item after substitution.

However, when the CDDL definition needs to explicitly identify the positions where substitutions can occur, the typename defined in Figure 1 is recommended:

```
variable<varid> = #6.42(varid)
```

Figure 1: Recommended typenames for CDDL

4. IANA Considerations

RFC-Editor note: Please replace "42" throughout this document by the actual tag allocated and delete this note.

IANA is requested to allocate the tag 42 as in Table 1, with the present document as the specification reference.

+-----+-----+-----+-----+			
Tag	Data Item	Semantics	
+-----+-----+-----+-----+			
42	any	CBOR variable	
+-----+-----+-----+-----+			

Table 1: Values for Tags

5. Security Considerations

The security considerations of [RFC 7049](#) apply; the tag introduced here not expected to raise security considerations beyond those.

Obviously, any process for performing variable substitution as outlined in [Section 2](#) needs to ensure that all of its inputs are derived considering the security objectives, and that the inputs are actually intended to fit together for this substitution.

For example, if a signed value includes the variables to be substituted, but does not contain a unique identifier for the template, then the structure resulting in substituting the variables in the template must not be considered as having been signed. Identifying templates by a hash, or by an identifier (that needs to be specific to the actual value and version of the template) is recommended.

6. References

6.1. Normative References

- [I-D.greevenbosch-appsawg-cbor-cddl]
Birkholz, H., Vigano, C., and C. Bormann, "Concise data definition language (CDDL): a notational convention to express CBOR data structures", [draft-greevenbosch-appsawg-cbor-cddl-11](#) (work in progress), July 2017.
- [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>>.

6.2. Informative References

- [I-D.ietf-core-comi]
Veillette, M., Stok, P., Pelov, A., and A. Bierman, "CoAP Management Interface", [draft-ietf-core-comi-02](#) (work in progress), December 2017.
- [I-D.ietf-lpwan-coap-static-context-hc]
Minaburo, A. and L. Toutain, "LPWAN Static Context Header Compression (SCHC) for CoAP", [draft-ietf-lpwan-coap-static-context-hc-02](#) (work in progress), September 2017.
- [RFC7252] Shelby, Z., Hartke, K., and C. Bormann, "The Constrained Application Protocol (CoAP)", [RFC 7252](#), DOI 10.17487/RFC7252, June 2014, <<https://www.rfc-editor.org/info/rfc7252>>.

Contributors

Laurent Toutain suggested the creation of a mechanism for indicating which parts of a CBOR data item are not yet available or subject to change.

Jim Schaad contributed to the security considerations.

Acknowledgements

TBD

Author's Address

Carsten Bormann
Universitaet Bremen TZI
Postfach 330440
Bremen D-28359
Germany

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