Workgroup: Network Working Group
Internet-Draft:
draft-bormann-cbor-draft-numbers-00
Published: 22 February 2023
Intended Status: Informational
Expires: 26 August 2023
Authors: C. Bormann
Universität Bremen TZI

Managing CBOR numbers in Internet-Drafts

Abstract

CBOR-based protocols often make use of numbers allocated in a registry. While developing the protocols, those numbers may not yet be available. This impedes the generation of data models and examples that actually can be used by tools.

This short draft proposes a common way to handle these situations, without any changes to existing tools. Such changes are very well possible in the future, at which time this draft will be updated.

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 26 August 2023.

Table of Contents

1. Introduction

(Please see abstract.) [RFC8949]

2. The Problem

A CBOR-based protocol might want to define a structure using CDDL [RFC8610][RFC9165], like that in Figure 1 (based on [RFC9290]):

```
problem-details = {
  ? &(title: -1) => oltext
  ? &(detail: -2) => oltext
  ? &(instance: -3) => ~uri
  ? &(response-code: -4) => uint .size 1
  ? &(base-uri: -5) => ~uri
  ? &(base-lang: -6) => tag38-ltag
  ? &(base-rtl: -7) => tag38-direction
  / ... /
  * (uint .feature "extension") => any
}
```

Figure 1: CDDL data model, final form

The key numbers shown in this structure are likely to be intended for allocation in an IANA section.

The key numbers will be used in an example in the specification such as shown in Figure 2.

```
{
  / title /         -1: "title of the error",
  / detail /        -2: "detailed information about the error",
  / instance /      -3: "coaps://pd.example/FA317434",
  / response-code / -4: 128, / 4.00 /
  4711: {
     / ... /
  }
}
```

Figure 2: CBOR-diag example, final form

However, during development, these numbers are not yet fixed; they are likely to move around as parts of the specification are added or deleted.

3. The Anti-Pattern

What not to do during development:

```
problem-details = {
  ? "title" => oltext
  ? "detail" => oltext
  ? "instance" => ~uri
  ? "response-code" => uint .size 1
  ? "base-uri" => ~uri
  ? "base-lang" => tag38-ltag
  ? "base-rtl" => tag38-direction
  / ... /
  * (uint .feature "extension") => any
}
```

Figure 3: CDDL data model, muddled form

```
{
  "title": "title of the error",
  "detail": "detailed information about the error",
  "instance-code": "coaps://pd.example/FA317434",
  "response-code": 128, / 4.00 /
  4711: {
     / ... /
  }
}
```

Figure 4: CBOR-diag example, muddled form

This makes the model and the examples compile/check out without allocating numbers, but it also leads to several problems:

- It becomes hard to assess what the storage/transmission cost of these structures will be.

- What is being checked in the CI (continuous integration) for the document is rather different from the final form.

- Draft implementations trying to make use of these provisional structures have to cater for text strings, which may not actually be needed in the final form (which might expose specification bugs once numbers are used, too late in the process).

- The work needed to put in the actual numbers, once allocated, is significant and error-prone.

- It is not certain the CI system used during development can interact with the RFC editor's way of editing the document for publication.

4. What to do during spec development

To make the transition to a published document easier, the document is instead written with the convention demonstrated in the following:

```
problem-details = {
  ? &(title-CPA: -1) => oltext
  ? &(detail-CPA: -2) => oltext
  ? &(instance-CPA: -3) => ~uri
  ? &(response-code-CPA: -4) => uint .size 1
  ? &(base-uri-CPA: -5) => ~uri
  ? &(base-lang-CPA: -6) => tag38-ltag
  ? &(base-rtl-CPA: -7) => tag38-direction
  / ... /
  * (uint .feature "extension") => any
}
```

Figure 5: CDDL data model, development form

CPA is short for "code point allocation", and is a reliable search key for finding the places that need to be updated after allocation.An earlier concept for this draft used TBD in place of CPA, as do many draft specifications being worked on today. TBD is better recognized than CPA, but also could be misunderstood to mean further work by the spec developer is required. A document submitted for publications should not really have "TBD" in it.

In the IANA section, the table to go into the registry is prepared as follows:

Key value	Name	CDDL Type	Brief description	Reference
CPA-1	title	text / tag38	short, human-readable summary of the problem shape	RFC XXXX
CPA-2	detail	text / tag38	human-readable explanation specific to this occurrence of the problem	RFC XXXX
CPA-3	instance	~uri	URI reference identifying specific occurrence of the problem	RFC XXXX
CPA-4	response-code	uint .size 1	CoAP response code	RFC XXXX
CPA-5	base-uri	~uri	Base URI	RFC XXXX
CPA-6	base-lang	tag38-ltag	Base language tag (see tag38)	RFC XXXX
CPA-7	base-rtl	tag38-direction	Base writing direction (see tag38)	RFC XXXX

Table 1: IANA table, development form

The provisionally made up key numbers will then be used in an example in the specification such as:

```
{
  / title-CPA /         -1: "title of the error",
  / detail-CPA /        -2: "detailed information about the error",
  / instance-CPA /      -3: "coaps://pd.example/FA317434",
  / response-code-CPA / -4: 128, / 4.00 /
  4711: {
     / ... /
  }
}
```

Figure 6: CBOR-diag example, development form

A "removeInRFC" note in the draft points the RFC editor to the present document so the RFC editor knows what needs to be done at which point. In the publication process, it is easy to remove the -CPA suffixes and CPA prefixes for the RFC editor while filling in the actual IANA allocated numbers and removing the note.

5. IANA Considerations

This document makes no requests of IANA. However, it specifies a procedure that can be followed during draft development that has a specific role for IANA and the interaction between RFC editor and IANA at important points during this development. This procedure is intended to be as little of an onus as possible, but that is the author's assessment only. IANA feedback is therefore requested.

6. Security considerations

The security considerations of [RFC8610] and [RFC8949] apply.

7. References

7.1. Normative References

[RFC8610] Birkholz, H., Vigano, C., and C. Bormann, "Concise Data Definition Language (CDDL): A Notational Convention to Express Concise Binary Object Representation (CBOR) and JSON Data Structures", RFC 8610, DOI 10.17487/RFC8610, June 2019, <https://www.rfc-editor.org/rfc/rfc8610>.

[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>.

[RFC9165] Bormann, C., "Additional Control Operators for the Concise Data Definition Language (CDDL)", RFC 9165, DOI 10.17487/RFC9165, December 2021, <https://www.rfc-editor.org/rfc/rfc9165>.

7.2. Informative References

[RFC9290] Fossati, T. and C. Bormann, "Concise Problem Details for Constrained Application Protocol (CoAP) APIs", RFC 9290, DOI 10.17487/RFC9290, October 2022, <https://www.rfc-editor.org/rfc/rfc9290>.

Acknowledgements

This document was motivated by the AUTH48 experience for RFC 9200..RFC 9203. Then, Jaime Jiménez made me finally write this document. Marco Tiloca provided useful comments on an early presentation of this idea.

Author's Address

Carsten Bormann

Universität Bremen TZI
Postfach 330440
D-28359 Bremen
Germany

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