

2017-01-09: CBOR WG

- Concise Binary Object Representation
Maintenance and Extensions
 1. Formal process: Take RFC 7049 to IETF STD level
(October 2018 milestone)
 2. Standardize CDDL as a data definition language
(May 2018 milestone, actual: August 2018)
 3. (Maybe define a few more CBOR tags, as needed.)

CDDL

Henk Birkholz, Christoph Vigano, [Carsten Bormann](#)
draft-ietf-cbor-cddl

CDDL



draft-ietf-cbor-cddl-
→ RFC editor queue



2019-03-25



Peeking post-1.0

- SUIT people tell us they'd now really like:
 - Import function (here: for COSE)
 - Namespace control (related to import)
- At some point, a module registry may make sense
- (For more ideas, see also IETF102 slides)

Free hugs?

CDDL Doctors?

- OPS/MGT has SNMP doctors and YANG doctors
- CDDL should empower the individual designer
- Still, some form of tutoring and coaching is useful
 - `cddl.space` — watch this space...

CBOR (RFC 7049) bis

Concise Binary Object Representation

Carsten Bormann, 2019-03-27

Levels of Errors

- (not) well-formed — CBOR Syntax
 - Error: Not recoverable (outside diagnostic tools)
 - See also Appendix C (pseudocode)
- (not) valid — CBOR Semantics
 - Error: Presentable to the application in principle
- (not) expected —
Application Syntax and Semantics
 - This is often expressed in CDDL

#17: Well-formed/valid

- Mostly integrated; not yet rebased
- Remaining items mostly delete redundant text; need to check it really is redundant

To do: strict

- A strict decoder only accepts preferred encoding
 - Again, this also has an application component
- Similar: deterministic-checking decoder
- Text about security miracles already toned down

#18: Tag validity

- Clarifies some of the tag validity provisions in 7049
- Does not question the direction taken by 7049

Tag validity

- Example: Tag 1 (POSIX time) takes int/float
- Maybe should have taken decimal as well (then we may not have needed Tag 1001)
- Similar: Tag 36 (mime message) only takes UTF-8
Should have taken byte string as well
Now have 257 for that.

Reactionary Tag Validity

- Tag is defined with a certain set of substructures (structural compatibility)
- A new substructure can never accede to an existing Tag
- There is little ambiguity about Tag validity

Progressive Tag Validity

- Tag is defined with abstract semantics
- Any substructure that fulfills that abstract semantics will do
- E.g., Tag 1 could take any number in \mathbb{R}
- E.g., Tag expecting array of numbers could take typed array (Tag 64..87)

Application expectedness of Tags

- CDDL: #6.36(tstr) vs. #6.36(tstr/bstr)
- Note that standard prelude says:
mime-message = #6.36(tstr)
- But application saying #6.36(tstr/bstr) is unambiguously using the tag

Ways forward

- Clarify the reactionary tag validity approach taken in RFC 7049 (done well by PR #18)
 - Much stricter
 - Still modulated by application expectedness
- Move to progressive tag validity
 - Much more flexible
 - Potential interoperability surprises outside CDDL

How to specify Tag type system

- New tag definition should document
 - expectations from tagged value (e.g., $\in \mathbb{R}$)
 - Abstract “type” of the result

Other todos

- Check Strict some more
- Clean up preferred encoding; base deterministic encoding on this
- Slightly Update IANA considerations
 - (We have another specification required in 1+1)
- One more round of reviews, and then WGLC?

Other CBOR housekeeping

draft-bormann-cbor- sequence

- Patterned after RFC 7464 (JSON Sequences)
 - Format definition, Media type, Content-Format, ...
 - But quite different:
 - CBOR is easy to concatenate (no ASCII RS needed)
 - No attempt at error recovery needed or possible
- People already want to put normative references to this into their documents

CBOR tag definitions

Carsten Bormann, 2018-07-17

Batteries included

- RFC 7049 predefines 18 Tags
 - Time, big numbers (bigint, float, decimal), various converter helpers, URI, MIME message
- Easy to register your own CBOR Tags
 - > 20 more tags: 6 for COSE; UUIDs, Sets, binary MIME, Perl support, language tagged string, compression

Status of Tags drafts

- **OID**: On charter, kitchen sink, expired. Needs work.
- **Array**: On charter, WGLC completed, new I-D needed.
- **Time**: Off charter; solved for now by FCFS registration (3-byte tag 1001); move spec to RFC how?
- **Template**: Off charter (will likely be done with SCHC anyway)
- **“Useful tags”**: Maybe document some of the more useful registered tags in an RFC on its own (could include Time)?

Draft-ietf-cbor-array-tags-03 (was draft-jroatch-cbor-tags)

- Provide tags for homogeneous arrays represented in byte strings

uint	sint	float
uint8	sint8	binary16
uint16	sint16	binary32
uint32	sint32	binary64
uint64	sint64	binary128

- Inspired by JavaScript
- 12×2: Both LSB and MSB first
- Reserves 24 contiguous tags
- Provides a tag for other homogeneous arrays
- Provides a tag for multidimensional arrays

Work needed on array-tags

- Lots of good editorial suggestions (#2–5, 8, 10)
- Add use case explanations for homogeneous and clamped arrays
- Explain that BE vs. LE (and what precision) is an application decision here; no implications on deterministic encoding