

CBOR @ 2023-10-04

time-tag • edn-literals • update-8610-grammar

draft-ietf-cbor-time-tag

Submitted to IESG 2023-08-23

AD review of -09: 2023-10-03

Quick response: 2023-10-03

Pull request: <https://github.com/cbor-wg/time-tag/pull/19>

1. Are we OK with these changes?
2. Maybe also change **registry group** name (next slide)

time-tag new registries

1. (registers tag numbers 1001 to 1003 in cbor-tags registry/group)
2. creates a registry "Timescales"
3. creates a registry "Map Keys"
4. creates a registry **group** "CBOR Time Tag Parameters" for 2 and 3
Vision: New Tags create new registry groups

Question: Create a **common** registry group "CBOR Tag Parameters"?
→ can be used for other registries that Tag definitions might want to create

Maybe use **existing** registry group "CBOR Tags"?
→ create "Timescales" and "Time Tag Map Keys" registry right in there?

CBOR-associated languages

- **CBOR** = representation and interchange format (binary, concise, efficient)
 - low-level visualization in text as **cbor-pretty** (hex with comments)

Two associated textual languages:

- **EDN** (cbor-diag) → examples, diagnostics
 - Text form for single **instance** (item/sequence), convert back and forth (**cbor.me**)
 - Derived from **JSON**, made more useful for humans, added binary, tags, ...
- **CDDL** → specification, validation
 - Describe specific data **model** (grammar)
 - Inspired by **ABNF**, can describe JSON, CBOR, CSV*

draft-ietf-cbor-edn-literals

Originally: Define EDN literals

Now really: EDN maintenance and extensions

- Changes in -04 OK?
 - #8 End-of-line comments (with #)
 - #7 Hex floating point
 - #6 Document when a number is floating point vs. integer value
 - Example for fractional time in dt ''
 - Unicode editorial and ABNF fixes
 - More editorial fixes

<https://github.com/cbor-wg/edn-literal/issues>

- #11 encoding indicators
 - encoding indicators well defined, but little implementation experience
 - (not in current ABNF, nor in edn-abnf implementation)
 - new: encoding indicators for app-prefix data (dt ' ')?
- #12 IANA considerations for app-prefix registry
 - mostly steal from JSONPath function extensions?

JSONPath "function

extensions"

3.2. Function Extensions

This specification defines a new "Function Extensions sub-registry" in a new "JSONPath Parameters registry", with the policy "expert review" ([Section 4.5](#) of [\[BCP26\]](#)).

The experts are instructed to be frugal in the allocation of function extension names that are suggestive of generally applicable semantics, keeping them in reserve for functions that are likely to enjoy wide use and can make good use of their conciseness. The expert is also instructed to direct the registrant to provide a specification ([Section 4.6](#) of [\[BCP26\]](#)), but can make exceptions, for instance when a specification is not available at the time of registration but is likely forthcoming. If the expert becomes aware of function extensions that are deployed and in use, they may also initiate a registration on their own if they deem such a registration can avert potential future collisions.

EDN Plan

- address #11 and #12 this week, -05
- WGLC (with update-8610-grammar, see below)

draft-ietf-cbor-update-8610-grammar

- RFC 8610 (CDDL) errata and maintenance
- author believes this is ready for WGLC
 - (comments leading to a -01 now also appreciated)
- Proposal: WGLC this in parallel with EDN-literals
 - both have non-trivial amounts of **ABNF**
 - helps to look at them in a comparative way;
see also Appendix B of EDN-literal for a starter