draft-ietf-cbor-cddl-control-02

Carsten Bormann, CBOR IETF110 meeting, 2021-03-08
New control operators for CDDL: .cat/.plus, .abnf/.abnfb, .feature

Was waiting for implementation experience with .abnf/.abnfb —
- ABNF is hard — thoughts about splitting it off
- ABNF finally implemented in early March, no more waiting needed
- One new observation: ABNF is indentation-sensitive

Proposal: go ahead, but add .det for “dedenting cat” (PR #6) ➔ .plus/.cat/.det, .abnf/.abnfb, .feature
Proposed next steps

• (Status: Implemented in CDDL tool 0.8.21)
• ➔ Submitted trial balloon –03 with .det
• ? Get pre-WGLC reviews
• ➔ WGLC
draft-ietf-cbor-packed-02

- draft-ietf-cbor-packed-02:
  - Add Tag allocations for suffix packing
  - Add WoT example
- Missing: more details for table setup (and its extensibility)
- Issue: add circumfix packing? (compare draft-bormann-lpwan-cbor-template-02)
- **Next step**: Get the implementation out, further implementation work
- Heads-up for next thing: **Streaming** packing? (Can be added to this later.)
draft-bormann-cbor-time-tag-04

- Tag 1001 has been around for a while and is in active use
- Add timescale key (UTC ~ NTP, TAI ~ GPS ~ PTP)
- Add clock quality (~ PTP [RFC8575], uncertainty [GUM] and guarantees [RFC3161])
- Add duration (delta, 1002), period (interval, 1003)

- Next steps:
  - ? Could live on as individual for more time (registrations have been made)
  - ? Could become WG document with a view of completion in 2021
• Intention: hitchhikers’ guide for the CBOR tag landscape
• Collect descriptive information that explains
  • Why sometimes there are multiple tags for one area of application, and
  • What distinguishes them
  • How to select between them
• Slow progress ➔ any day now [please send PRs!]
• Plan is to snapshot and publish after some time (2022?)
• Also *defines* some tags (e.g., 63 for encoded cbor seq, 65535 etc. for invalid tags)
• Essentially unchanged for a year now
  • last change: extracted computed literals and .abnf ➔ -control
• More controls (less urgent needs): .pcre, endianness, bitfields
• Alternative representations: CDDL in JSON (interface between tools!)
• Language extensions:
  • Module superstructure (export/import), namespaces
  • Co-occurrence constraints (hard to do right; piggyback on what?)
  • Literal syntaxes (regexes, literals for tags) (waiting for strong use cases)
Essentially unchanged for a year now
  last change: extracted computed literals and .abnf ➔ -control✔

More controls (less urgent needs): .pcre, endianness, bitfields

Alternative representations: CDDL in JSON (interface between tools!)
  ➔ could extract for informational

Language extensions:
  Module superstructure (export/import), namespacing ➔ highest perceived need
  Co-occurrence constraints (hard to do right; piggyback on what?)
  Literal syntaxes (regexes, literals for tags) (waiting for strong use cases)
ai=28 (128-bit integers/floats)?
• Extension beyond 64-bit arguments (int/float, not meaningful for the rest)
• Can use tags to handle some of this now:
  • Tag 2/3 for uint128, nint128 (use one more byte, but can compact leading 0)
  • Tag 5 can handle float128 and more, but only with icky conversion work
  • Tag 83 is for arrays of float128, but can send one of them as well
• Should we define specific tags (float128, float256???)?
• Or should we open Pandora’s ai=28 box?