

Packed CBOR

draft-ietf-cbor-packed-01

Carsten Bormann, CBOR interim, 2021-01-27

-01: What has changed?

- Trivial and editorial stuff, more robust terminology
- **Clear split** between unpacking itself (referencing) and table setup
- Add **suffix** packing (with prefix packing, this is now affix packing)
- Specify **container** affixes (array, map)
- Call the one table setup tag defined “**basic packed CBOR**”
 - Discuss application-specific and explicitly tagged additional setup

Issues due to suffix packing

- Now three tables in setup: item sharing, prefixes, suffixes
 - All are arrays of data items, indexed by unsigned integers
- Naïvely, this would **double** the affix tags allocation used
- We don't have a corpus, but:
 - **Hunch:** prefix packing is more important than suffix packing
 - Allocate more to prefix than to suffix?
 - Requires different arithmetic between them
 - Tag 6 use for prefix already creates a small difference

Issues due to container affixes

- Container affixes are pretty much obvious for arrays
- Map affixes:
 - Maps are unordered, so prefix \equiv suffix? ;??!?
 - How to handle keys that are in both affix map and rump map?
 - -01 Proposal: **override**
 - Prefix: rump overrides prefix
 - Suffix: suffix overrides rump
 - Note: No removal of entries provided

-00: efficient Item and Prefix references

- Item references: 16 simple values (1+0), one single-byte Tag → 48+512+131072 (1+1, 1+2, 1+4)
- Prefix references: Reuse tag; use more tags (32+4096+268435456) Do the same (but not necessarily the same sizes) separately for suffix
- Total reservation: 4/7 simple values, 1 1+0 tag (1/24), 1/8 1+1, 1/16 1+2, ...
- Worth it if we think this will be a widely used part of CBOR
- Could be less aggressive and less efficient, but why?

–01 Strawman: prepend in table setup

Building tables from multiple sources

- A table setup “applies” to any data item
- Default: Three empty tables (nothing to reference)
 - Changed by application environment, or explicitly by a tag
- Basic-packed-CBOR: tag **prepends** enclosed tables to tables that already apply to it
 - Could come from application environment: Static dictionary use case
 - Could come from some tag enclosing a structure that contains this tag (“composition”)
- Note: tables enclosed in tag are already using prepended numbering (!)
- Basic-packed-CBOR doesn’t use bucket structure, but other ones could

How to build tables

- Position in table is relevant
 - At least within a **bucket**:
 - Items: 16, 48, 512, 131072
 - Prefixes/**Suffixes**: 32, 4096, 268435456
- Combining imported and locally defined tables
 - Use imported only? Use locally defined only (= -00)?
 - When using both, sequence becomes important when a bucket overflows

Next steps

- Complete current discussion
- Progress this draft with only a minimal setup tag (\approx basic-packed-CBOR)
 - We can always define more later (e.g., based on `ni` : URI scheme, defining a table setup format media type in the process)
- In parallel: Discuss usage in specific application environments
 - Define profile (subset, e.g., shared items only)?
 - Any setup tags specific to the application environment?
 - Static dictionary defined by application environment (e.g., media type)?