

# draft-ietf-cbor-serialization

Status and discussion, Interim January 21 2026

Laurence Lundblade, CBOR Interim Meeting January 21, 2026

# Draft Status — PRs merged

- No PRs merged since last interim meeting on January 7, 2026
- 3 minor PR merged since draft -01
  - dCBOR reference
  - Add Ken Takayama as contributor
  - Small clarification for deterministic serialization

# Draft Status — Same 16 Open Issues

- Request for sub section on when to use general — has open PR
- General recommendations section — has open PR
- Consensus on tag 2 and 3 equivalence #30 — **has open PR**
- Get consensus on name “ordinary serialization”
- Request to make some appendices normative
- Rohan requests additional security considerations; I’ve requested Rohan write them
- “SHOULD” related to big numbers — either justify or make into MUSTS — **has open PR**
- TODO: test vectors
- TODO: rename “.det” control operator
- TODO: deterministic encoding of tag 0
- Shortest length for wrapping byte strings — I don’t support this, will probably turn into WONT FIX
- 5 issues Anders filed that need resolution on list or response from him

**PRs opened since last interim meeting**

“TODO” means writing work is next step, not discussion, consensus or review

# Draft Status — Open PRs

- Big number SHOULDs — Simple PR says decoders MUST ignore leading zeros and MUST treat an empty string as zero
- Big number appendicies — 1) Explain data model modifications defined for big numbers. 2) Suggestion implementation strategies.
- Rohan's General Review: many comments about many parts of the document — processing and resolution is in progress
- Add when-to-use-general serialization section — Ready for review
- Add general recommendations section — Discussion subject for this interim

Will publish draft -02 when  
more PRs are merged

**Discussion:**

**Is ordinary serialization SHOULD, MUST or just defined**

# Summarized Recommendations from PR

- CBOR Libraries (e.g., Ciborium, QCBOR, tinyCBOR, ...)
  - SHOULD support ordinary serialization
  - MAY support deterministic serialization
- Future protocol designs (e.g., EAR, merkle-tree-proofs... )
  - SHOULD say that ordinary serialization SHOULD be used
- Protocol implementations (fully coded & deployed CWT, SenML, EAT,...)
  - SHOULD implement ordinary serialization

# Pros and Cons of Recommendations

- Preferred serialization has been relatively successful
  - Commonly implemented; no major interop problems reported
  - But, lack of clarity for definite-lengths, NaN payloads, big numbers
  - Still some confusion among protocol designers
- SHOULD support Ordinary
  - Remove much of the confusion; provide clear direction
  - Almost trivial and sensible transition because of widespread support of preferred serialization
- MUST support Ordinary
  - Very clear for ecosystem, but mandates unnecessary work for some use cases
- Say nothing, just define ordinary serialization
  - Increased confusion between ordinary and preferred
  - Missed opportunity for clarity