CBOR Working group
Internet-Draft
Intended status: Informational
Expires: 3 March 2026

C. Bormann
Universitt Bremen TZI

30 August 2025

CDDL 2.0 and beyond — a draft plan
draft-bormann-cbor-cddl-2-draft-07

Abstract

The Concise Data Definition Language (CDDL) today is defined by RFC 8610, RFC 9165, RFC 9682, and RFC\ 9741). RFC 9165 and the latter (as well as some more application specific specifications such as RFC 9090) have used the extension point provided in RFC 8610, the control operator.

As CDDL is used in larger projects, feature requirements become known that cannot be easily mapped into this single extension point. Hence, there is a need for evolution of the base CDDL specification itself.

The present document provides a roadmap towards a "CDDL 2.0"; it is intended to serve as a basis for implementations that evolve with the concept of CDDL 2.0. It is based on draft-bormann-cbor-cddl-freezer, but is more selective in what potential features it takes up and more detailed in their discussion. This document is intended to evolve over time; it might spawn specific documents and then retire, or it might eventually be published as a roadmap document.

About This Document

This note is to be removed before publishing as an RFC.

Status information for this document may be found at https://datatracker.ietf.org/doc/draft-bormann-cbor-cddl-2-draft/.

Discussion of this document takes place on the cbor Working Group mailing list (mailto:cbor@ietf.org), which is archived at https://mailarchive.ietf.org/arch/browse/cbor/. Subscribe at https://www.ietf.org/mailman/listinfo/cbor/.

Source for this draft and an issue tracker can be found at https://github.com/cbor-wg/cddl-2.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

(Please see abstract.)

Note that the existing extension point can be exercised for new features in parallel to the work described here. [RFC9741] forms part of the first set of specifications going forward from the CDDL-2 project together with [RFC9682].

The rest of this introduction gives a rough overview over what could be the development plan for CDDL 1.1, 2.0, 2.5.

1.1. CDDL 1.1 + 2 plan (standards track)

This section documents the status in Summer 2024.

CDDL 1.1 milestone (documents technically complete, implemented):

* "CDDL 1.1": [RFC9682], _Grammar_ fixes: Empty files (enabling CDDL 2), non-literal tags, errata fixes. Approved document, in RFC editor queue (EDIT state) at the time of writing.

* Parallel to CDDL 1.1: More _control_ operators [RFC9741]: Additional control operators, another iteration like RFC 9165 before.

CDDL 2.0 work:

* Technically complete before *IETF 119*: CDDL 2.0: [I-D.ietf-cbor-cddl-modules] (import/include directives, implemented). Feedback is available from IETF 119, one open technical issue (sockets); WGLC 1H2025.

* Potentially, further directives to be added. No proposals are ripe for specification; this work could go into a second document constituting "CDDL 2.1" so we have the well-understood import/include available now.

"CDDL 2.5":

* Being prepared in *1H2025*: CDDL 2.5: Section 3 of the present document ("_annotations_", plus some functionality enabled by that). The requirements are reasonably well-understood; the specific form this takes needs to be worked out. Enables, e.g., Section 5 of [I-D.bormann-cbor-cddl-freezer] (co-occurrence).

1.2. Other documents

Not on the main line of development, but important ancillary work:

* (Informational, implemented): Section 6 (alternative representations) of [I-D.bormann-cbor-cddl-freezer]: CDDL-in-JSON format(s) for interchange of CDDL model information between tools.

* (Informational, companion to [I-D.ietf-cbor-cddl-modules]): [I-D.bormann-cbor-rfc-cddl-models] (builds standard collection of referenceable models).

* (BCP? Informational?): [I-D.bormann-cbor-draft-numbers] (BCP for handling assigned numbers during draft stage; can stay informational as the work described is completed and any reference to the document erased before a specification using it would be published).

* Application-oriented literal e'' [I-D.ietf-cbor-edn-e-ref] makes use of [I-D.ietf-cbor-edn-literals] so that diagnostic notation can refer to named numbers that are specified in CDDL. Implemented, see [enum-literals] for an introduction.

More explorative at this point:

* (Standards-Track?) The remaining Section 2 of this document: application-oriented literals in CDDL mirroring the work in [I-D.ietf-cbor-edn-literals].

* (Informational or Standards-Track?): [I-D.bormann-cbor-cddl-csv] (using CDDL to model CSV documents).

Important CBOR work that may be reflected in some CDDL extensions:

* Evolving Extended Diagnostic Notation [I-D.ietf-cbor-edn-literals]. While EDN and CDDL are independent languages (with EDN rooted in JSON and CDDL in ABNF and Relax-NG), they are often used together, and developments in one may spawn parallel work in the other.

* Common Deterministic Encoding (CDE) [I-D.ietf-cbor-cde] and related documents. These do define CDDL operators already, which may be sufficient for initial use; this might be extended once more experience has been gained.

* Packed CBOR [I-D.ietf-cbor-packed]. CDDL already can be used to describe the original data item represented in a packed data item. Requirements for describing the latter have not yet been collected; there is some relation to transformation (Section 3.2) that might need to be explored.

2. Mending syntax deficits

The previous content of this section formed the basis for [RFC9682], except for Section 2.1.

2.1. Tag-oriented Literals

Incomplete, see Appendix A.1.

3. Processing model: Beyond Validation

Proposal Status: experiments with implementations ongoing
Compatibility: backwards compatible

The basic (implicit) processing model for CDDL 1.0 applies a CDDL data model to a data item and returns a Boolean that indicates whether the data item matches that model ("_validation_").

Section 4 of [RFC9165] extends this model with named "_features_". A validation can indicate which features were used. Validation could also be parameterized with information about what features are allowed to be used, enabling variants (see Section 4 of [RFC9165] and [useful] for examples).

3.1. Annotations

The cddl tool (Appendix F of [RFC8610]) also supports experimental forms of "annotating" a validated data item with information about which rules were used to support validation, currently entirely based on the information that is in a standard CDDL 1.0 data model. This leads to a more general concept of "_annotation_", where the data model specification supports "annotating" the validated instance by optionally supplying information in the model. (The annotated result is a special case of a "post-schema validation instance" [PSVI], here one where the data item itself is only augmented, not changed, by the process.)

Annotations could in turn provide input to further validation steps, as is often done with Schematron validation in Relax-NG; with an appropriate evaluation language this can be used for checking co-occurrence constraints (Section 5 of [I-D.bormann-cbor-cddl-freezer]).

3.2. Transformation

Finally, annotations are a first step to _transformation_, i.e., describing how a validated data item should be interpreted as a transformed data item by performing certain computations. This generally requires even more support from an evaluation language, simple transformations such as adding in default values may not need much support though.

3.3. Next Steps

At this time, existing experimental implementations do not lead to a clear choice for what processing model enhancements should be in CDDL 2.0 follow-ons. This document proposes to continue the experimentation and document good approaches.

4. Module superstructure

The previous content of this section formed the basis for [I-D.ietf-cbor-cddl-modules]. Additional work might be started on the ideas outlined in the subsections of this section.

4.1. Cross-universe references

See Appendix A.2.

4.2. ABNF is a lot like CDDL

Many of the constructs defined here for CDDL also could be used with ABNF specifications. ABNF would definitely benefit from a standard way to import snippets from existing RFCs. Since CDDL contains ABNF support (Section 3 of [RFC9165]), it would be natural to make some of the functionality discussed in this section available for ABNF as well.

5. IANA Considerations

This document makes no requests of IANA.

6. Security considerations

The security considerations of [RFC8610] apply.

7. References

7.1. Normative References

[RFC8610] Birkholz, H., Vigano, C., and C. Bormann, "Concise Data Definition Language (CDDL): A Notational Convention to Express Concise Binary Object Representation (CBOR) and JSON Data Structures", RFC 8610, DOI 10.17487/RFC8610, June 2019, <https://www.rfc-editor.org/rfc/rfc8610>.

[RFC9165] Bormann, C., "Additional Control Operators for the Concise Data Definition Language (CDDL)", RFC 9165, DOI 10.17487/RFC9165, December 2021, <https://www.rfc-editor.org/rfc/rfc9165>.

7.2. Informative References

[enum-literals] "[Cbor] Getting diagnostic notation examples in drafts under control", 26 February 2024, <https://mailarchive.ietf.org/arch/msg/cbor/D8h_0Egog89GaRLFNwb1VfKlHI4>.

[EXTRACT-RB] "extract.rb — extract CDDL from an enum-style IANA registry", n.d., <https://github.com/cabo/common-cddl/blob/main/extract.rb>.

[I-D.bormann-cbor-cddl-csv] Bormann, C. and H. Birkholz, "Using CDDL for CSVs", Work in Progress, Internet-Draft, draft-bormann-cbor-cddl-csv-06, 3 March 2025, <https://datatracker.ietf.org/doc/html/draft-bormann-cbor-cddl-csv-06>.

[I-D.bormann-cbor-cddl-freezer] Bormann, C., "A feature freezer for the Concise Data Definition Language (CDDL)", Work in Progress, Internet-Draft, draft-bormann-cbor-cddl-freezer-15, 28 February 2025, <https://datatracker.ietf.org/doc/html/draft-bormann-cbor-cddl-freezer-15>.

[I-D.bormann-cbor-draft-numbers] Bormann, C., "Managing CBOR codepoints in Internet-Drafts", Work in Progress, Internet-Draft, draft-bormann-

cbor-draft-numbers-06, 7 July 2025,
<https://datatracker.ietf.org/doc/html/draft-bormann-cbor-draft-numbers-06>.

[I-D.bormann-cbor-rfc-cddl-models]
Bormann, C., "CDDL models for some existing RFCs", Work in Progress, Internet-Draft, draft-bormann-cbor-rfc-cddl-models-06, 29 August 2025,
<https://datatracker.ietf.org/doc/html/draft-bormann-cbor-rfc-cddl-models-06>.

[I-D.ietf-cbor-cddl-modules]
Bormann, C. and B. Moran, "CDDL Module Structure", Work in Progress, Internet-Draft, draft-ietf-cbor-cddl-modules-04, 3 March 2025, <https://datatracker.ietf.org/doc/html/draft-ietf-cbor-cddl-modules-04>.

[I-D.ietf-cbor-cde]
Bormann, C., "CBOR Common Deterministic Encoding (CDE)", Work in Progress, Internet-Draft, draft-ietf-cbor-cde-12, 7 July 2025, <https://datatracker.ietf.org/doc/html/draft-ietf-cbor-cde-12>.

[I-D.ietf-cbor-edn-e-ref]
Bormann, C., "External References to Values in CBOR Diagnostic Notation (EDN)", Work in Progress, Internet-Draft, draft-ietf-cbor-edn-e-ref-02, 2 July 2025,
<https://datatracker.ietf.org/doc/html/draft-ietf-cbor-edn-e-ref-02>.

[I-D.ietf-cbor-edn-literals]
Bormann, C., "CBOR Extended Diagnostic Notation (EDN)", Work in Progress, Internet-Draft, draft-ietf-cbor-edn-literals-18, 7 July 2025,
<https://datatracker.ietf.org/doc/html/draft-ietf-cbor-edn-literals-18>.

[I-D.ietf-cbor-packed]
Bormann, C. and M. Gtschow, "Packed CBOR", Work in Progress, Internet-Draft, draft-ietf-cbor-packed-16, 7 July 2025, <https://datatracker.ietf.org/doc/html/draft-ietf-cbor-packed-16>.

[PSVI] "Use Cases for XML Schema PSVI API", 24 June 2002,
<https://www.w3.org/XML/2002/05/psvi-use-cases>.

[RFC9682] Bormann, C., "Updates to the Concise Data Definition Language (CDDL) Grammar", RFC 9682, DOI 10.17487/RFC9682, November 2024, <https://www.rfc-editor.org/rfc/rfc9682>.

[RFC9741] Bormann, C., "Concise Data Definition Language (CDDL): Additional Control Operators for the Conversion and Processing of Text", RFC 9741, DOI 10.17487/RFC9741, March 2025, <https://www.rfc-editor.org/rfc/rfc9741>.

[useful] "Useful CDDL", n.d., <https://github.com/cbor-wg/cddl/wiki/Useful-CDDL>.

Appendix A. Fridge

This appendix contains sections that may not make it to a 2.0 milestone, but might be part of a followup.

A.1. Tag-oriented Literals

Proposal Status: rough idea, porting from EDN
Compatibility: backward (not forward)

Some CBOR tags often would be most natural to use in a CDDL spec with a literal syntax that is tailored to their semantics instead of the serialization of their tag content in CBOR. There is currently no way to add such syntaxes, no defined extension point either.

The specification "CBOR Extended Diagnostic Notation (EDN): Application-Oriented Literals, ABNF, and Media Type" [I-D.ietf-cbor-edn-literals] defines application-oriented literals, e.g., of the form

```
dt'2019-07-21T19:53Z'
```

for datetime items. With additional considerations for unambiguous syntax, a similar literal form could be included in CDDL.

This proposal opens a namespace for the prefix that indicates an application specific literal. A registry could be provided to turn this namespace into a genuine extension point. (This is currently the production bsqual in Appendix B of [RFC8610].)

The syntax provided in [I-D.ietf-cbor-edn-literals] does not enable the use of named CDDL rules — using it directly in CDDL would have the same flaw that is being fixed for tag numbers in Section 3.2 of [RFC9682].

A.2. Cross-universe references

Often, a CDDL specification needs to import from specifications in a different language or platform.

A.2.1. IANA references

In many cases, CDDL specifications make use of values that are specified in IANA registries. The proposed .iana control operator can be used to reference such a set of values.

The reference needs to be able to point to a draft, the registry of which has not been established yet, as well as to an established IANA registry.

An example of such a usage might be:

```
cose-algorithm = int .iana ["cose", "algorithms", "value"]
```

Unfortunately, the vocabulary employed in IANA registries has not been designed for machine references. In this case, the potential values would come from applying the XPath expression

```
//iana:registry[@id='algorithms']/iana:record/iana:value
```

to https://www.iana.org/assignments/cose/cose.xml, plus some filtering on the records returned that only leaves actual allocations. Section 3.1 of [I-D.bormann-cbor-rfc-cddl-models] contains an example of a CDDL module that is automatically generated from those assignments. (The code for this extraction is available in the document source repository of [I-D.bormann-cbor-rfc-cddl-models] as [EXTRACT-RB].)

Additional functionality may be needed for filtering with respect to other columns of the registry record, e.g., <capabilities> in the case of this example.

Acknowledgements

TBD

Author's Address

Carsten Bormann
Universitt Bremen TZI
Postfach 330440
D-28359 Bremen
Germany

Phone: +49-421-218-63921
Email: cabo@tzi.org