

Network Working Group
Internet-Draft
Intended status: Informational
Expires: July 31, 2018

C. Bormann
Universitaet Bremen TZI
January 27, 2018

**A feature freezer for the Concise Data Definition Language (CDDL)
draft-bormann-cddl-freezer-00**

Abstract

In defining the Concise Data Definition Language (CDDL), some features have turned up that would be nice to have. In the interest of completing this specification in a timely manner, the present document collects nice-to-have features that did not make it into the first RFC for CDDL.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of [BCP 78](#) and [BCP 79](#).

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on July 31, 2018.

Copyright Notice

Copyright (c) 2018 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to [BCP 78](#) and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

- [1. Introduction](#) [2](#)
- [2. Cuts](#) [2](#)
- [3. Literal syntax](#) [2](#)
 - [3.1. Computed Literals](#) [2](#)
 - [3.2. Tag-oriented Literals](#) [3](#)
 - [3.3. Regular Expression Literals](#) [3](#)
- [4. Embedded ANBF](#) [3](#)
- [5. Controls](#) [3](#)
 - [5.1. Control operator .pcre](#) [4](#)
- [6. Module superstructure](#) [4](#)
- [7. IANA Considerations](#) [4](#)
- [8. Security considerations](#) [4](#)
- [9. References](#) [4](#)
 - [9.1. Normative References](#) [4](#)
 - [9.2. Informative References](#) [5](#)
- Acknowledgements [5](#)
- Author's Address [5](#)

1. Introduction

(TO DO: Insert an extended form of the abstract first here, expanding the reference to [[I-D.ietf-cbor-cddl](#)].)

There is always a danger for a document like this to become a shopping list; the intention is to develop this document further based on real-world experience with the first CDDL standard.

2. Cuts

Section 3.5.3 of [[I-D.ietf-cbor-cddl](#)] alludes to a new language feature, `_cuts_`, and defines it in a fashion that is rather focused on a single application in the context of maps and generating better diagnostic information about them.

The present document is expected to grow a more complete definition of cuts, with the expectation that it will be upwards-compatible to the existing one in [[I-D.ietf-cbor-cddl](#)], before this possibly becomes a mainline language feature in a future version of CDDL.

3. Literal syntax

3.1. Computed Literals

CDDL cannot compute.

This is a liability in at least two situations:

- o integers often need to be defined relative to an offset. It might be desirable to be able to write something like:

```
base = 16777216
a = base + 1
b = base + 2
```

- o some string literals (in particular, complex regular expressions) would best be composed from components. This could be done by adding a concatenation operator (maybe even "+" as used for addition above), or by adding string interpolation to the string literal syntax.

3.2. Tag-oriented Literals

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 their serialization in CBOR. There is currently no way to add such syntaxes, no defined extension point either.

3.3. Regular Expression Literals

Regular expressions currently are notated as strings in CDDL, with all the string escaping rules applied once. It might be convenient to have a more conventional literal format for regular expressions, possibly also providing a place to add modifiers such as "/i". This might also imply "text .regexp ...", which with the proposal in [Section 5.1](#) then raises the question of how to indicate the regular expression flavor.

4. Embedded ANBF

It would often be desirable to define a text string type by employing ABNF [[RFC5234](#)] [[RFC7405](#)] embedded into the CDDL specification. Currently, that ABNF would usually need to be translated into a regular expression (if that is even possible).

Note that some applications of computed literals for strings could be covered by such a feature (or partially vice versa).

5. Controls

Controls are the main extension point of the CDDL language. It is relatively painless to add controls to CDDL. Several candidates have been identified that aren't quite ready for adoption, of which one shall be listed here.

5.1. Control operator .pcre

There are many variants of regular expression languages. Section 3.8.3 of [[I-D.ietf-cbor-cddl](#)] defines the .regexp control, which is based on XSD [[XSD2](#)] regular expressions. As discussed in that section, the most desirable form of regular expressions in many cases is the family called "Perl-Compatible Regular Expressions" ([[PCRE](#)]); however, no formally stable definition of PCRE is available at this time for normatively referencing it from an RFC.

The present document defines the control operator .pcre, which is similar to .regexp, but uses PCRE2 regular expressions. More specifically, a ".pcre" control indicates that the text string given as a target needs to match the PCRE regular expression given as a value in the control type, where that regular expression is anchored on both sides. (If anchoring is not desired for a side, "." needs to be inserted there.)

6. Module superstructure

CDDL rules could be packaged as modules and referenced from other modules. There could be some control of namespace pollution, as well as unambiguous referencing ("versioning").

This is probably best achieved by a pragma-like syntax which could be carried in CDDL comments, leaving each module to be valid CDDL (if missing some rule definitions to be imported).

7. IANA Considerations

This document makes no requests of IANA.

8. Security considerations

The security considerations of [[I-D.ietf-cbor-cddl](#)] apply.

9. References

9.1. Normative References

[[I-D.ietf-cbor-cddl](#)]
Birkholz, H., Vigano, C., and C. Bormann, "Concise data definition language (CDDL): a notational convention to express CBOR data structures", [draft-ietf-cbor-cddl-00](#) (work in progress), July 2017.

9.2. Informative References

- [PCRE] "Perl-compatible Regular Expressions (revised API: PCRE2)", n.d., <<http://pcre.org/current/doc/html/>>.
- [RFC5234] Crocker, D., Ed. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", STD 68, [RFC 5234](#), DOI 10.17487/RFC5234, January 2008, <<https://www.rfc-editor.org/info/rfc5234>>.
- [RFC7405] Kyzivat, P., "Case-Sensitive String Support in ABNF", [RFC 7405](#), DOI 10.17487/RFC7405, December 2014, <<https://www.rfc-editor.org/info/rfc7405>>.
- [XSD2] Biron, P. and A. Malhotra, "XML Schema Part 2: Datatypes Second Edition", World Wide Web Consortium Recommendation REC-xmlschema-2-20041028, October 2004, <<http://www.w3.org/TR/2004/REC-xmlschema-2-20041028>>.

Acknowledgements

Many people have asked for CDDL to be completed, soon. These are usually also the people who have brought up observations that led to the proposals discussed here. Sean Leonard has campaigned for a regexp syntax.

Author's Address

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

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

