

T2TRG
Internet-Draft
Intended status: Informational
Expires: 1 September 2022

C. Bormann, Ed.
Universität Bremen TZI
28 February 2022

Semantic Definition Format (SDF) for Data and Interactions of Things:
Compact Notation
draft-bormann-asdf-sdf-compact-02

Abstract

The Semantic Definition Format (SDF) is a format for domain experts to use in the creation and maintenance of data and interaction models in the Internet of Things. It was created as a common language for use in the development of the One Data Model liaison organization (OneDM) definitions. Tools convert this format to database formats and other serializations as needed.

The SDF format is mainly intended for interchange between machine generation and machine processing. However, there is often a need for humans to look at and edit SDF models.

Similar to the way Relax-NG as defined in ISO/IEC 19757-2 has an XML format and a compact format (Annex C), this specification defines a compact format to go along SDF's JSON format.

The present version of this document is mostly a proof of concept, but was deemed useful to obtain initial feedback on the approach taken.

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-asdf-sdf-compact/>.

Discussion of this document takes place on the A Semantic Definition Format for Data and Interactions of Things (asdf) Working Group mailing list (<mailto:asdf@ietf.org>), which is archived at <https://mailarchive.ietf.org/arch/browse/asdf/>.

Status of This Memo

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

Internet-Draft

OneDM SDF compact

February 2022

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 1 September 2022.

Copyright Notice

Copyright (c) 2022 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 Revised BSD License text as described in Section 4.e of the [Trust Legal Provisions](#) and are provided without warranty as described in the Revised BSD License.

Table of Contents

- [1.](#) Introduction [2](#)
- [2.](#) Overview [3](#)
 - [2.1.](#) Example Definition [4](#)
- [3.](#) References [5](#)
 - [3.1.](#) Normative References [5](#)
 - [3.2.](#) Informative References [5](#)
- [Appendix A.](#) Example in SDF JSON format [5](#)
- Acknowledgements [7](#)
- Author's Address [7](#)

[1.](#) Introduction

The Semantic Definition Format (SDF) is a format for domain experts to use in the creation and maintenance of data and interaction models in the Internet of Things. It was created as a common language for

use in the development of the One Data Model liaison organization (OneDM) definitions. Tools convert this format to database formats and other serializations as needed.

The SDF format is mainly intended for interchange between machine generation and machine processing. However, there is often a need for humans to look at and edit SDF models.

Similar to the way Relax-NG as defined in ISO/IEC 19757-2 has an XML format and a compact format (Annex C), this specification defines a compact format to go along SDF's JSON format.

The present version of this document is mostly a proof of concept, but was deemed useful to obtain initial feedback on the approach taken.

(References: [[I-D.ietf-asdf-sdf](#)], [[RELAXNG](#)].)

The intention is to be able to bidirectionally translate between compact and JSON form, without appreciable semantic losses. This will allow viewing SDF in compact form, apply edits if needed, and then continuing processing it in JSON form. As a limitation of this approach, it will be difficult to always recreate the order of map entries (members of JSON objects) in the JSON form; this order is essentially arbitrary as maps (JSON objects) in JSON are unordered. (In the long run, it may be useful to define a canonical order in the SDF specification or here.)

An initial prototype of a converter from compact form to JSON form was constructed during the ASDF/WISHI hackathon preceding IETF110. A more complete, bidirectionally operating version of this tool is planned for release soon.

[2.](#) Overview

The SDF compact format is a YAML file [[YAML](#)]; a good part of the work needed for a compact representation is already done by the increased user-friendliness of YAML over JSON.

In addition, all the sections defined with `named<...>` in the CDDL

definition of SDF have been compacted into map entries with space-separated keys, giving the kind first and the name next. This saves the need for another level of hierarchy and reminds the reader of the kind of item being specified.

The map key description is replaced by `:`, which also is rendered in the most compact form possible in YAML.

The dataqualities readable, writable, and observable, together with optionality, are compressed into a four-character map key: The first three are translated into rwo when set (default in SDF) and into --- when not set. The fourth character is `?` for optional and `!` for required.

The value of this abbreviated key is a CDDL [[RFC8610](#)] rendition of the attributes defined in the jsonschema production in [Appendix A](#) of [[I-D.ietf-asdf-sdf](#)]. To further reduce noise, a top-level array production in the CDDL can be represented as an array in the YAML (i.e., does not require additional quotes).

[2.1.](#) Example Definition

This is an SDF-compact version of a slightly modified copy of the `sdfobject-cadence.sdf.json` model found at the time of writing in OneDM's SDF playground. (This example was chosen more or less randomly; better examples can probably be found. The modification is the addition of a unit quality.)

info:

```
copyright: Copyright 2018-2019 Open Connectivity Foundation, Inc. All rights reserved.
version: '2019-06-11'
title: Cadence
license: https://github.com/one-data-model/oneDM/blob/master/LICENSE
```

object cadence:

```
:: This Resource describes the cadence, which is the number of
  revolutions of crank per minute when cyclists pedal the
```

pedals. The unit, which is the default unit, is rpm. The cadence Property is a read-only value that is provided by the server. When range (from "oic. r. baseresource") is omitted the default is 0 to +MAXFLOAT.

property cadence:
r-o!: integer .ge 0
unit: 1/min
:: This Property describes the rate at which a cyclist is pedalling/turning the pedals.

property range:
r-o?: [2*2 integer]
:: The valid range for the Property in the Resource as an integer. The first value in the array is the minimum value, the second value in the array is the maximum value.

property step:
r-o?: integer
:: Step value across the defined range when the range is an integer. This is the increment for valid values across the range; so if range is 0..10 and step is 2 then valid values are 0,2,4,6,8,10.

The result of automatically converting this YAML file using the prototype `sdfc` tool back into the JSON form of SDF is given in [Appendix A](#). Except for the unit addition, it is semantically identical to the `sdfobject-cadence.sdf.json`. Differences are visible in the order of map entries (members in JSON objects); a future version of the `sdfc` tool could attempt to preserve more of this order, even though it does not carry semantics.

[3.](#) References

[3.1.](#) Normative References

[I-D.ietf-asdf-sdf]

Koster, M. and C. Bormann, "Semantic Definition Format (SDF) for Data and Interactions of Things", Work in Progress, Internet-Draft, [draft-ietf-asdf-sdf-10](#), 16 January 2022, <<https://www.ietf.org/archive/id/draft-ietf-asdf-sdf-10.txt>>.

[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](https://www.rfc-editor.org/info/rfc8610), DOI 10.17487/RFC8610, June 2019, <<https://www.rfc-editor.org/info/rfc8610>>.

3.2. Informative References

- [RELAXNG] ISO/IEC, "Information technology – Document Schema Definition Language (DSDL) – Part 2: Regular-grammar-based validation – RELAX NG", ISO/IEC 19757-2, 15 December 2008.
- [YAML] Ben-Kiki, O., Evans, C., and I. dot Net, "YAML Ain't Markup Language (YAML™) version 1.2", Revision 1.2.2, 1 October 2021, <<https://yaml.org/spec/1.2.2/>>.

[Appendix A](#). Example in SDF JSON format

===== NOTE: '\ ' line wrapping per [RFC 8792](#) =====

```
{
  "info": {
    "title": "Cadence",
    "license": "https://github.com/one-data-model/oneDM/blob/master/\
              LICENSE",
    "version": "2019-06-11",
    "copyright": "Copyright 2018-2019 Open Connectivity Foundation, \
                Inc. All rights reserved."
  },
```

```
"sdfObject": {
  "cadence": {
    "description": "This Resource describes the cadence, which is \
the number of revolutions of crank per minute when cyclists pedal \
the pedals. The unit, which is the default unit, is rpm. The \
cadence Property is a read-only value that is provided by the server \
. When range (from \"oic. r. baseresource\") is omitted the default \
is 0 to +MAXFLOAT.",
    "sdfProperty": {
      "step": {
        "type": "integer",
        "writable": false,
        "description": "Step value across the defined range when \
the range is an integer. This is the increment for valid values \
across the range; so if range is 0..10 and step is 2 then valid \
```

values are 0,2,4,6,8,10."

```
    },
    "range": {
      "type": "array",
      "items": {
        "type": "integer"
      },
      "maxItems": 2,
      "minItems": 2,
      "writable": false,
      "description": "The valid range for the Property in the \
Resource as an integer. The first value in the array is the minimum \
value, the second value in the array is the maximum value."
    },
    "cadence": {
      "type": "integer",
      "minimum": 0,
      "writable": false,
      "description": "This Property describes the rate at which \
a cyclist is pedalling/turning the pedals."
    }
  },
  "sdfRequired": [
    "#/sdfObject/cadence/sdfProperty/cadence"
  ]
}
}
```

Acknowledgements

The idea for this draft originated at the IETF110 ASDF/WISHI hackathon. The author would like to thank the attendees for initial feedback.

Author's Address

Carsten Bormann (editor)
Universität Bremen TZI
Postfach 330440
D-28359 Bremen
Germany
Phone: +49-421-218-63921
Email: cabo@tzi.org