

ASDF  
Internet-Draft  
Intended status: Standards Track  
Expires: 11 May 2022

C. Bormann, Ed.  
Universität Bremen TZI  
J. Romann  
Universität Bremen  
7 November 2021

Semantic Definition Format (SDF): Mapping files  
draft-bormann-asdf-sdf-mapping-00

## 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.

An SDF specification often needs to be augmented by additional information that is specific to its use in a particular ecosystem or application. SDF mapping files provide a mechanism to represent this augmentation.

## Discussion Venues

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

Discussion of this document takes place on the asdf Working Group mailing list ([asdf@ietf.org](mailto:asdf@ietf.org)), which is archived at <https://mailarchive.ietf.org/arch/browse/asdf/>.

Source for this draft and an issue tracker can be found at <https://github.com/cabo/sdf-mapping>.

## 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-Draft

SDF Mapping

November 2021

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 11 May 2022.

## Copyright Notice

Copyright (c) 2021 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

<a href="#">1.</a>	Introduction . . . . .	<a href="#">2</a>
<a href="#">1.1.</a>	Terminology and Conventions . . . . .	<a href="#">3</a>
<a href="#">2.</a>	Overview . . . . .	<a href="#">3</a>
<a href="#">2.1.</a>	Example Definition 1 (ecosystem: IPSO/OMA) . . . . .	<a href="#">3</a>
<a href="#">2.2.</a>	Example Definition 2 (ecosystem: W3C WoT) . . . . .	<a href="#">4</a>
<a href="#">3.</a>	Formal Syntax of SDF mapping files . . . . .	<a href="#">6</a>
<a href="#">4.</a>	IANA Considerations . . . . .	<a href="#">7</a>
<a href="#">4.1.</a>	Media Type . . . . .	<a href="#">7</a>
<a href="#">4.2.</a>	Registries . . . . .	<a href="#">8</a>
<a href="#">5.</a>	Security Considerations . . . . .	<a href="#">8</a>
<a href="#">6.</a>	References . . . . .	<a href="#">8</a>
<a href="#">6.1.</a>	Normative References . . . . .	<a href="#">8</a>
<a href="#">6.2.</a>	Informative References . . . . .	<a href="#">9</a>
	Acknowledgements . . . . .	<a href="#">9</a>
	Authors' Addresses . . . . .	<a href="#">9</a>

## [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.

An SDF specification often needs to be augmented by additional information that is specific to its use in a particular ecosystem or application. SDF mapping files provide a mechanism to represent this augmentation.

### [1.1](#). Terminology and Conventions

The definitions of [\[I-D.ietf-asdf-sdf\]](#) apply.

The term "byte" is used in its now-customary sense as a synonym for "octet".

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [BCP 14](#) [\[RFC2119\]](#) [\[RFC8174\]](#) when, and only when, they appear in all capitals, as shown here.

## [2](#). Overview

An SDF mapping file provides augmentation information for one or more SDF definitions. Its main contents is a map from SDF name references (Section 4.3 of [\[I-D.ietf-asdf-sdf\]](#)) to a set of qualities.

When processing the mapping file together with one or more SDF definitions, these qualities are added to the SDF definition at the referenced name, as in a merge-patch operation [\[RFC7396\]](#). Note that this is somewhat similar to the way sdfRef (Section 4.4 of [\[I-D.ietf-asdf-sdf\]](#)) works, but in a mapping file the arrows point in the inverse direction (from the augments to the augmented).

### [2.1](#). Example Definition 1 (ecosystem: IPSO/OMA)

An example for an SDF mapping file is given in Figure 1. This mapping file is meant to attach to an SDF specification published by OneDM, and to add qualities relevant to the IPSO/OMA ecosystem.

```
// Note that this example uses namespaces to identify elements of
// the referenced specification(s), but has un-namespaced quality
// names. These two kinds of namespaces are probably unrelated, and
// we may need to add quality namespacing to SDF (independent of a
// potential feature to add namespace references to definitions that
// are not intended to go into the default namespace -- these are SDF
// definition namespaces and not quality namespaces, which are one
// meta-level higher).
```

\* Start of mapping file for certain OneDM playground models:

```
{
  "info": {
    "title": "IPSO ID mapping"
  },
  "namespace": {
    "onedm": "https://onedm.org/models"
  },
  "defaultNamespace": "onedm",
  "map": {
    "#/sdfObject/Digital_Input": {
      "id": 3200
    },
    "#/sdfObject/Digital_Input/sdfProperty/Digital_Input_State": {
      "id": 5500
    },
    "#/sdfObject/Digital_Input/sdfProperty/Digital_Input_Counter": {
      "id": 5501
    }
  }
}
```

Figure 1: A simple example of an SDF mapping file

## [2.2.](#) Example Definition 2 (ecosystem: W3C WoT)

This example shows a the translation of a hypothetical W3C WoT Thing Model into an SDF model plus a mapping file to catch Thing Model attributes that don't currently have SDF qualities defined.

// The example probably would be more useful with, say, protocol

```
// bindings. This is left for a future version of this example, and/  
// or a future specification that specifically addresses how to map  
// Thing Models into SDF. (There is also the separate requirement to  
// transform a Thing Description into the kind of information that  
// can be represented in SDF plus instance information, such as IP  
// addresses or specific node names.) Finally, namespaces are all  
// wrong in this example.
```

\* The input: WoT Thing Model

```
{  
  "@context": ["http://www.w3.org/ns/td"],  
  "@type" : "tm:ThingModel",  
  "title": "Lamp Thing Model",  
  "titles": {  
    "en": "Lamp Thing Model",  
    "de": "Thing Model für eine Lampe"  
  },  
  "properties": {  
    "status": {  
      "description": "Current status of the lamp",  
      "descriptions": {  
        "en": "Current status of the lamp",  
        "de": "Aktueller Status der Lampe"  
      },  
      "type": "string",  
      "readOnly": true  
    }  
  }  
}
```

Figure 2: Input: WoT Thing Model

\* The output: SDF model

```
{
  "info": {
    "title": "Lamp Thing Model"
  },
  "namespaces": {
    "wot": "http://www.w3.org/ns/td"
  },
  "defaultNamespace": "wot",
  "sdfObject": {
    "LampThingModel": {
      "label": "Lamp Thing Model",
      "sdfProperty": {
        "status": {
          "description": "Current status of the lamp",
          "writable": false,
          "type": "string"
        }
      }
    }
  }
}
```

Figure 3: Output 1: SDF Model

\* The other output: SDF mapping file

```
{
  "info": {
    "title": "Lamp Thing Model: WoT TM mapping"
  },
  "namespace": {
    "wot": "http://www.w3.org/ns/td"
  },
  "defaultNamespace": "wot",
  "map": {
    "#/sdfObject/LampThingModel": {
      "titles": {
        "en": "Lamp Thing Model",
        "de": "Thing Model für eine Lampe"
      }
    }
  }
}
```

```

    }
  },
  "#/sdf0bject/LampThingModel/sdfProperty/status": {
    "descriptions": {
      "en": "Current status of the lamp",
      "de": "Aktueller Status der Lampe"
    }
  }
}
}
}

```

Figure 4: Output 2: SDF Mapping File

### 3. Formal Syntax of SDF mapping files

An SDF mapping file has three optional components that are taken unchanged from SDF: The info block, the namespace declaration, and the default namespace. The mandatory fourth component, the "map", contains the mappings from a SDF name reference (usually a namespace and a JSON pointer) to a nested map providing a set of qualities to be merged in at the site identified in the name reference.

Figure 5 describes the syntax of SDF mapping files using CDDL [\[RFC8610\]](#).

```

start = sdf-mapping

```

```

sdf-mapping = {
  ? info: sdfinfo                ; This will be required in most process po
  ? namespace: named<text>
  ? defaultNamespace: text
  ? map: { * sdf-pointer => additionalqualities}
}

```

```

; we can't really be much more specific here:
additionalqualities = named<any>

; ----- import from SDF:

sdfinfo = {
  ? title: text
  ? version: text
  ? copyright: text
  ? license: text
  EXTENSION-POINT<"info-ext">
}

; Shortcut for a map that gives names to instances of X (has text keys and v
named<X> = { * text => X }

EXTENSION-POINT<f> = ( * (text .feature f) => any ) ; only used in framework

sdf-pointer = text ; .regexp curie-regexp -- TO DO!

```

Figure 5: CDDL definition of SDF mapping file

## [4.](#) IANA Considerations

### [4.1.](#) Media Type

IANA is requested to add the following Media-Type to the "Media Types" registry.

Name	Template	Reference
sdf-mapping+json	application/sdf-mapping+json	RFC XXXX, <a href="#">Section 4.1</a>

Table 1: A media type for SDF mapping files



// remove this note.

Type name: application  
Subtype name: sdf-mapping+json  
Required parameters: none  
Optional parameters: none  
Encoding considerations: binary (JSON is UTF-8-encoded text)  
Security considerations: [Section 5](#) of RFC XXXX  
Interoperability considerations: none  
Published specification: [Section 4.1](#) of RFC XXXX  
Applications that use this media type: Tools for data and interaction modeling in the Internet of Things  
Fragment identifier considerations: A JSON Pointer fragment identifier may be used, as defined in [Section 6 of \[RFC6901\]](#).  
Person & email address to contact for further information: ASDF WG mailing list (asdf@ietf.org), or IETF Applications and Real-Time Area (art@ietf.org)  
Intended usage: COMMON  
Restrictions on usage: none  
Author/Change controller: IETF  
Provisional registration: no

## [4.2.](#) Registries

(TBD: After future additions, check if we need any.)

## [5.](#) Security Considerations

Some wider issues are discussed in [\[RFC8576\]](#).

(Specifics: TBD.)

## [6.](#) References

### [6.1.](#) 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-08](#), 25 October 2021, <<https://www.ietf.org/archive/id/draft-ietf-asdf-sdf-08.txt>>.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.

- [RFC6901] Bryan, P., Ed., Zyp, K., and M. Nottingham, Ed., "JavaScript Object Notation (JSON) Pointer", [RFC 6901](#), DOI 10.17487/RFC6901, April 2013, <<https://www.rfc-editor.org/info/rfc6901>>.
- [RFC7396] Hoffman, P. and J. Snell, "JSON Merge Patch", [RFC 7396](#), DOI 10.17487/RFC7396, October 2014, <<https://www.rfc-editor.org/info/rfc7396>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in [RFC 2119](#) Key Words", [BCP 14](#), [RFC 8174](#), DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.
- [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/info/rfc8610>>.

## [6.2](#). Informative References

- [RFC8576] Garcia-Morchon, O., Kumar, S., and M. Sethi, "Internet of Things (IoT) Security: State of the Art and Challenges", [RFC 8576](#), DOI 10.17487/RFC8576, April 2019, <<https://www.rfc-editor.org/info/rfc8576>>.

## Acknowledgements

This draft is based on discussions in the Thing-to-Thing Research Group (T2TRG) and the SDF working group. Input for [Section 2.1](#) was provided by Ari Keränen.

## Authors' Addresses

Carsten Bormann (editor)  
Universität Bremen TZI  
Postfach 330440  
D-28359 Bremen  
Germany

Phone: +49-421-218-63921  
Email: [cabo@tzi.org](mailto:cabo@tzi.org)

Jan Romann  
Universität Bremen

Email: [jan.romann@uni-bremen.de](mailto:jan.romann@uni-bremen.de)