# Citation File Format (CFF)

Specification - Version 1.0.0-beta

Stephan Druskat (mail@sdruskat.net)

19 September 2017

**Abstract**

The *Citation File Format* (*CFF*) is a human- *and* machine-readable format for citation files, which provide references to (research and scientific) software to be used for citation and other types of reference. The format aims to support all use cases for software citation described in [1]. CFF is serialized in YAML 1.2, and is therefore Unicode-based and cross-language (in terms of both natural language scripts and programming languages). This specification, together with the Unicode standard for characters, aims to provide all the information necessary to understand CFF, and to use (i.e., write) and re-use (i.e., read, validate, convert from) it. The specification is maintained openly at https://github.com/sdruskat/citation-file-format.

## Contents

**Introduction** **2**
- Status of this document . . . 2
- Rationale . . . 2
- Goals . . . 2
- Concepts . . . 2

**Format** **2**
- File structure . . . 2
- Formatting . . . 2
- Keys . . . 2
  - Details . . . 4
- Entities . . . 5
  - Roles . . . 6
- Statuses . . . 6
- Work Types . . . 7
- Schema . . . 8
- Examples . . . 8

**Infrastructure** **9**
- Creating CFF `CITATION` files . . . 9
- Reading CFF `CITATION` files . . . 9
- Validating CFF `CITATION` files . . . 9
- Converting CFF `CITATION` files . . . 9

**Notes** **9**

**License** **9**

**References** **9**

# Introduction

## Status of this document

This document reflects the first version of the *Citation File Format* (CFF). CFF has been developed in the context of the *Workshop on Sustainable Software for Science: Practice and Experiences (WSSSPE5.1)*, which was held on 6 September 2017 in Manchester, UK. More specifically, the constraints for CFF has been developed in the discusion and speed blogging group "Development and implementation of a standard format for CITATION files", whose members were Stephan Druskat (Humboldt-Universität zu Berlin, Germany), Neil Chue Hong (Software Sustainability Institute, University of Edinburgh, UK), Raniere Silva (Software Sustainability Institute, University of Manchester, UK), Radovan Bast (University of Tromsø, Norway), Andrew Rowley (University of Manchester, UK), and Alexander Konovalov (University of St. Andrews, UK).

CFF Version 1.0 has been developed by Stephan Druskat with contributions from the following.

CFF has been developed to provide the first iteration of a format for `CITATION` files which could be recommended to readers of the blog post which has been produced by the group during the workshop and shortly after, and which will be published on the blog page of the Software Sustainability Institute.

## Rationale

- Implement enablement for principles

## Goals

Implement the principles from "Software Citation Principles".

All-purpose citation format

## Concepts

- All keys can be used for all types although tooling (esp. conversion) may not take the into account
- Sections are part of the whole (e.g., book section is in book)

# Format

- ALSO CHECK AGAINST SOFT CIT IMPLEMENTATION WG GITHUB which is meant to support implementers
- Check TAGS (e.g., yaml.org/type/) - custom tag repository?
- how to reference custom schema?

## File structure

## Formatting

key: whitespace value

## Keys

CFF defines the following keys.

Table 1: Complete list of CFF keys.

| **CFF Key** | **CFF Data Type** | **CFF Description** |
|---|---|---|
| abbreviation | String | The abbreviation of the work |
| abstract | String | The abstract of a work |
| authors | Collection of **entities** | The author of a work |
| bibtex | String | A BibTeX version of the reference |
| collection-title | String | The title of a collection or proceedings |
| collection-type | String | The type of a collection |
| commit | String | The (e.g., Git) commit hash or (e.g., Subversion) revision number of the work |
| conference | Entity | The conference where the work was presented |
| contact | Collection of **entities** | The contact person for a work |
| copyright | String | The copyright information pertaining to the work |
| data-type | String | The data type of a data set |
| database | String | The name of the database where a work was accessed/is stored |
| database-provider | Entity | The provider of the database where a work was accessed/is stored |
| date-accessed | Date | The date the work has been last accessed |
| date-downloaded | Date | The date the work has been downloaded |
| date-published | Date | The date the work has been published |
| date-released | Date | The date the work has been released |
| department | String | The department where a work has been produced |
| dependencies | String (*URI*) | A Uniform Resource Identifier pointing to a resource that makes the dependencies of a work accessible |
| doi | String | The DOI of the work |
| edition | String | The edition of the work |
| editors | Collection of **entities** | The editors of a work |
| editors-series | Collection of **entities** | The editors of a series in which a work has been published |
| entry | String | An entry in the collection that constitutes the work |
| filename | String | The name of the electronic file containing the work |
| format | String | The format in which a work is represented |
| institution | Entity | The institution where a work has been produced or published |
| isbn | String | The ISBN of the work |
| issn | String | The ISSN of the work |
| issue | Integer | The issue of a periodical in which a work appeared |
| issue-date | String | The publication date of the issue of a periodical in which a work appeared |
| issue-title | String | The name of the issue of a periodical in which the work appeared |
| journal | String | The name of the journal/magazine/newspaper/periodical where the work was published |
| keywords | Collection of strings | Keywords pertaining to the work |
| languages | Collection of strings | The language of the work |
| license | String | The license under which a work is licensed |
| license-url | String (*URL*) | The URL of the license text under which a work is licensed |
| loc-start | Integer | The line of code in the file where the work starts |
| loc-end | Integer | The line of code in the file where the work ends |
| message | String | A message providing the user with instructions on how to cite the work the `CITATION` file is attached to |
| month | Integer | The month in which a work has been published |
| nihmsid | String | The NIHMSID of a work |
| notes | String | Notes pertaining to the work |
| number | String | The accession number for a work |

| CFF Key | CFF Data Type | CFF Description |
|---|---|---|
| number-volumes | Integer | The number of volumes making up the collection in which the work has been published |
| pages | Integer | The number of pages of the work |
| patent-states | String | The states for which a patent is granted |
| pmcid | String | The PMCID of a work |
| publisher | Entity | The name of the publisher who has published the work |
| recipients | Collection of **entities** | The recipient of a personal communication |
| repository | String (*URL*) | The repository where the work is stored |
| repository-code | String (*URL*) | The version control system where the source code of the work is stored |
| repository-artifact | String (*URL*) | The repository where the (executable/binary) artifact of the work is stored |
| section | String | The section of a work that is referenced |
| sender | Collection of **entities** | The sender of a personal communication |
| status | **Status string** | The publication status of the work |
| start | Integer | The start page of the work |
| thesis-type | String | The type of the thesis that is the work |
| title | String | The title of the work |
| translators | Collection of **entities** | The translator of a work |
| type | **Work Type string** | The type of the work |
| url | String (*URL*) | The URL of the work |
| version | String | The version of the work |
| volume | Integer | The volume of the periodical in which a work appeared |
| volume-title | String | The title of the volume in which the work appeared |
| year | Integer | The year in which a work has been published |
| year-original | Integer | The year of the original publication |

### Details

This section details some of the keys where necessary to avoid ambiguity.

**abstract**

- If the work is a journal paper or other academic work: The abstract of the work.
- If the work is a film, broadcast or similar: The synopsis of the work.

**department**

- If the work is a thesis: The academic department where the thesis has been produced.
- If the work is a government document: The governmental department which has issued the document.

**dependencies**

- If the work is a software: A URL of a metadate entry, e.g., `http://depsy.org/package/python/nltk`; the URI or a URL of a file listing the software's dependencies, e.g., `file:///NOTICE` (if the artifact of the software version available from `repository-artifact` includes a `NOTICE` file in the root folder), or `https://github.com/user/project/blob/master/NOTICE`.

**format**

- If the work is a music file: The digital format in which a musical piece is saved, e.g., MP3.
- If the work is a data set: The digital format in which the data set is saved.
- If the work is a painting: The format of the painting, e.g., the width and height of the canvas.

**institution**

- If the work is a report: The institution where the report has been produced.
- If the work is a case: The court where a case has been held.
- If the work is a blog post: The institution responsible for running the blog.
- If the work is a patent, legal rule or similar: The issuing institution of the patent/rule.
- If the work is a grant: The funding agency sponsoring the grant.
- If the work is a thesis: The university where a thesis has been produced.
- If the work is a statute: The institution or geographical unit which the statute adheres to.
- If the work is a historical work, illuminated manuscript or similar: The library or archive where the work is held.
- If the work is a conference: The organisation which held the conference.

**languages**

- If the work is a book: The language in which the book is written.
- If the work is a software: The programming/markup languages in which the software is written.

**month**

- If the work is a conference: The month in which the conference has been held.
- If the work is a magazine article: The month in which the magazine issue containing the article has been published.

**number**

- If the work is a conference paper: E.g., the submission number of the paper
- If the work is a grant: The grant number provided by the funding agency.
- If the work is a work of art: E.g., the catalogue number provided by a museum holding the artwork.
- If the work is a report: The report number of a report.
- If the work is a patent: The patent number of the work.
- If the work is a historical work, illuminated manuscript or similar: The codex or folio number of a manuscript, or the library identifier for a manuscript.

**term**

- If the work is a dictionary or encyclopedia: The term in the dictionary or encyclopedia that is being referenced.

**title**

- If the work is a case: The name of the case (e.g., Name v. Name).

**version**

- If the work is a software: The version of the referenced software.

## Entities

Entity objects can represent different types of entities, e.g., a person, publishing company, or conference. In CFF, they are realized as collections with a defined set of keys. Only the key `name` is mandatory. When the entity represents a person, the `name` key must be formatted following the pattern `"{last names} :: {first names} : {middle names}"`. This pattern is used to parse names correctly, and implicitly disambiguate person entities from other entities. Therefore, if a non-person entity name follows this pattern, it must be given as `{first part of the name} \:: {second part of the name} \: {third part of the name}`.

Note that the whitespaces preceding and following the separators (`::`, `:`, `\::`, `\:`) are optional.

Table 2: Complete list of entity keys.

| Entity key | Entity Data Type | optional |
|---|---|---|
| name | String | |
| city | String | • |
| country | String | • |
| street | String | • |

| Entity key | Entity Data Type | optional |
|---|---|---|
| orcid | String | • |
| email | String | • |
| affiliation | String | • |
| tel | String | • |
| fax | String | • |
| website | String (*URL*) | • |
| date-start | Date | • |
| date-end | Date | • |
| location | String | • |
| role | **Role string** | • |

### Roles

An entity representing a person can be assigned a role. The defined roles are:

Table 3: Defined roles for entities.

| Key |
|---|
| **artist** |
| **assignee** (e.g., of a patent) |
| **benchmarker** (e.g., of a software) |
| **cartographer** |
| **composer** |
| **contributor** |
| **creator** |
| **designer** |
| **director** (e.g., of a movie) |
| **editor** (e.g., of an edited book/edition) |
| **evangelist** (e.g., for a software) |
| **insitution** (e.g., issuing a standard) |
| **inventor** |
| **manager** (e.g., of a software project) |
| **programmer** |
| **reporter** (e.g., of a court case) |
| **reporter** (e.g., of a software bug) |
| **researcher** (e.g., authoring a data set) |
| **software engineer** (e.g., for a software) |
| **technical writer** (e.g., of a software documentation) |
| **tester** (e.g., of a software) |
| **trainer** |

## Statuses

Works can have a different status of publication, e.g., journal papers. CFF provides the following defined statuses for works.

Table 4: Defined statuses for works

| Status (String) | Description |
|---|---|
| **in-preparation** | A work in preparation, e.g., a manuscript |
| **abstract** | The abstract of a work |
| **submitted** | A work that has been submitted for publication |

| Status (String) | Description |
|---|---|
| **in-press** | A work that has been accepted for publication but has not yet been published |
| **advance-online** | A work that has been published online in advance of publication in the target medium |

## Work Types

Table 5: Complete list of CFF work types.

| Work Type string |
|---|
| **art** |
| **article** |
| **audiovisual** |
| **bill** |
| **bill** |
| **blog** |
| **book** |
| **catalogue** |
| **conference** |
| **conference-paper** |
| **data** |
| **database** |
| **dictionary** |
| **edited-work** |
| **encyclopedia** |
| **film-broadcast - Film or Broadcast 1** |
| **generic** |
| **government-document** |
| **grant** |
| **hearing** |
| **historical-work** |
| **legal-case** |
| **legal-rule** |
| **magazine-article** |
| **manual** |
| **map** |
| **multimedia** |
| **music** |
| **newspaper-article** |
| **pamphlet** |
| **patent** |
| **personal-communication** |
| **proceedings** |
| **report** |
| **serial** |
| **slides** |
| **software** |
| **software-code** |
| **software-container** |
| **software-executable** |
| **software-virtual-machine** |
| **sound-recording** |
| **standard** |
| **statute** |
| **thesis** |

| Work Type string |
|---|
| **unpublished** |
| **video** |
| **website** |

## Schema

- Define one! (PyKwalify?)

## Examples

```yaml
- message: "If you use this software, please cite the software itself and the journal paper describing its
- TYPE: "SOFTWARE (Use YAML explicit typing? !software)"
  authors:
    - firstname: Stephan
      lastname: Druskat
      orcid: 1234-5678-9012-3456
    - firstname: Neil
      lastname: "Chue Hong"
    - firstname: Radovan
      lastname: Bast
      orcid: 1234-5678-9012-3456
  csv: "git://github.com/sdruskat/cffp"
  doi: 10043/zenodo.1234
  title: "Citation File Format Parser"
  version: "1.1.2"
- TYPE: JOURNAL
  authors:
    - firstname: Stephan
      lastname: Druskat
      orcid: 1234-5678-9012-3456
    - firstname: Neil
      lastname: "Chue Hong"
    - firstname: Radovan
      lastname: Bast2
      orcid: 1234-5678-9012-3456
  day: 9
  doi: 10043/zenodo.12345
  frompage: 1
  issue: 11
  journal: "Journal for Open Research Software (JORS)"
  month: september
  subtitle: "Version 1.0"
  title: "The Citation File Format"
  topage: 34
  volume: 2017
  year: 2017
```

# Infrastructure

### Creating CFF `CITATION` files

### Reading CFF `CITATION` files

- Use cases in software, cf. https://www.software.ac.uk/blog/2014-07-30-oh-research-software-how-shalt-i-cite-thee

### Validating CFF `CITATION` files

### Converting CFF `CITATION` files

# Notes

- Virtual machines? UID?
- Containers (docker)? UID?
- Active instances?

# License



# References

[1] A. M. Smith, D. S. Katz, K. E. Niemeyer, and FORCE11 Software Citation Working Group, "Software citation principles," *PeerJ Computer Science*, vol. 2, p. e86, Sep. 2016 [Online]. Available: https://doi.org/10.7717/peerj-cs.86