import-citation-cff: Import CFF metadata into LaTeX

Jisoo Song[a,b,*]

[a]Institute of Chemical Process, Seoul National University, Republic of Korea
[b]Department of Chemical and Biological Engineering, Seoul National University, Republic of Korea

Version v0.3.0 (2026-11-08)

Abstract

Modern research software and datasets often include a CFF (Citation File Format) file to provide metadata for proper citation. When you write the manuscript that is to be cited, you need to ensure that the manuscript and the CFF file have consistent metadata. import-citation-cff helps with this process by reading the metadata from the CFF file and incorporating it into the LaTeX document.

Contents

1 Introduction **2**

2 Requirements **2**

3 Usage **3**
3.1 Specifying the source level . . . 4
3.2 Reading CFF fields . . . 4
3.3 Iterating over authors . . . 4
3.4 Multiple affiliations . . . 4
3.5 Author style . . . 4
3.6 Text in the metadata . . . 5
3.7 Recording the CFF file . . . 5
3.8 Checking the title page . . . 5
3.9 Expanding metadata . . . 5

4 Built-in author styles **6**
4.1 `standard` . . . 6
4.2 `authblk` . . . 6
4.3 `rsc` . . . 6

*E-mail: jeesoo9595@snu.ac.kr

5 Author style plugins **7**
5.1 Writing a definition file 7
5.2 Plugin keys 8
5.3 Rendering order 8
5.4 Checking the title page 9
5.5 Supporting `cfflatexpand` 9

6 Reference **10**

7 Errors and warnings **13**

8 License **14**

1 Introduction

Software is a product of research and should be cited like one [1]. The Citation File Format (CFF) [2] makes this practical by keeping a YAML-format file with `cff` extension, typically named `CITATION.cff`, in the software repository. The CFF file holds the metadata needed to cite the software which an be read by both people and programs. This practice allows automated tools to easily discover and use the citation metadata.

The CFF file and the cited manuscript are often kept and developed together in a single repository, the latter usually written in LaTeX. Developers need to keep duplicate metadata in sync between the CFF file and the manuscript, which can be tedious and error-prone. This complication is even more pronounced when the software and the manuscript are developed in parallel, the former metadata being kept in the top level of the CFF file while the latter metadata being kept in the subsection of the CFF file such as `preferred-citation`.

import-citation-cff relieves this burden by keeping the CFF file as the single source of metadata and importing it into the LaTeX document. The metadata, such as the title, abstract, and authors, are imported as LaTeX macros that dynamically expand whenever the CFF file changes. This ensures that the metadata are synced across the repository, allowing for easier maintenance and automation of the development process.

2 Requirements

import-citation-cff parses the CFF file in Lua using the lt3luabridge and lua-tinyyaml packages. It needs either

- LuaLaTeX, which runs the Lua code directly, or
- pdfLaTeX with unrestricted shell escape (`--shell-escape`), which runs the Lua code with `texlua`.

It supports TeX Live 2025 and later.

3 Usage

This manuscript itself is an example of using import-citation-cff. Its `CITATION.cff` is

```
cff-version: 1.2.0
message: If you use import-citation-cff, please cite it using these metadata.
type: software
title: "import-citation-cff: Import CFF metadata into LaTeX"
abstract: >-
  Modern research software and datasets often include a CFF (Citation
  File Format) file to provide metadata for proper citation. When you
  write the manuscript that is to be cited, you need to ensure that the
  manuscript and the CFF file have consistent metadata.
  import-citation-cff helps with this process by reading the metadata
  from the CFF file and incorporating it into the LaTeX document.
authors:
  - family-names: Song
    given-names: Jisoo
    affiliation: >-
      Institute of Chemical Process,
      Seoul National University,
      Republic of Korea;
      Department of Chemical and Biological Engineering,
      Seoul National University,
      Republic of Korea
    email: "jeesoo9595@snu.ac.kr"
contact:
  - family-names: Song
    given-names: Jisoo
    email: "jeesoo9595@snu.ac.kr"
repository-code: https://github.com/JSS95/cfflatex
license: MIT
keywords:
  - LaTeX
  - Citation File Format
  - metadata
```

The LaTeX manuscript then imports and accesses the metadata by

```
\documentclass{l3doc}
\usepackage{import-citation-cff}
\includecff{CITATION.cff}
\title{\cff{title}}
\author{\cffauthor{1}{given-names}~\cffauthor{1}{family-names}}

\begin{document}
\maketitle
\begin{abstract}
  \cff{abstract}
\end{abstract}
\end{document}
```

3.1 Specifying the source level

By default, `\includecff` reads the top level of the CFF file. The `source=`⟨*key*⟩ option selects another citation under the top-level ⟨*key*⟩, such as `preferred-citation`. Any other top-level key is read in the same way. Note that CFF 1.2.0 defines `preferred-citation` as the only such key and allows no others, so a file with another key fails CFF validation.

3.2 Reading CFF fields

Once the CFF file is included, `\cff` reads any value of the citation by its path. The keys and the indices, counting from 1, that lead to it, are separated by slashes. `\cffcount` gives the length of a list. With the `CITATION.cff` of this manual,

`\cff{title}`	import-citation-cff: Import CFF metadata into LaTeX
`\cff{authors/1/family-names}`	Song
`\cff{keywords/2}`	Citation File Format
`\cffcount{keywords}`	3

The path is expanded first, so `\cff{authors/\cffauthorindex/email}` reads the email address of the current author in `\cffforeachauthor`.

3.3 Iterating over authors

`\cffforeachauthor` runs its argument once for each author of the citation, and `\cffauthorindex` gives the index of the current author. A document can thus list the authors without knowing how many there are. The following sets the author list of `\maketitle`, with `\and` between the authors:

```
\author{\cffforeachauthor{%
  \ifnum\cffauthorindex>1 \and\fi
  \cffauthor{\cffauthorindex}{given-names}~%
  \cffauthor{\cffauthorindex}{family-names}}}
```

3.4 Multiple affiliations

CFF gives each person one `affiliation` which is a nonempty string, and does not natively support multiple affiliations per author. import-citation-cff, allows multiple affiliations per author by separating them with semicolons in the `affiliation` field of the author. Affiliations of each author can be accessed individually by `\cffauthoraffiliation` and related commands. Refer to Section 6 for details.

3.5 Author style

`\cffsetauthors` declares the authors of the citation for the title page in an author style, which `\cffsetauthors[style=`⟨*name*⟩`]` selects. It marks each author with their affiliations, numbered among the distinct affiliations of the citation, and with a footnote for each contact of the citation that has the same names. Section 4 lists the styles that come with import-citation-cff, and Section 5 explains how to write others. The other keys of `\cffsetauthors` change how a name is made, how the authors and the affiliations are declared, and the marks and the footnotes (Section 6).

3.6 Text in the metadata

Values are inserted as text. The TEX special characters `\ { } % # $ & _ ~ ^` are printed as themselves, so LATEX markup in the CFF file is not interpreted. Line breaks inside a value become spaces, so the paragraphs of a multi-paragraph abstract run together, and spaces at either end are dropped. Non-ASCII characters are kept. With pdfLATEX, use a font encoding that has them, such as `T1`.

3.7 Recording the CFF file

`\includecff` records the CFF file as an input in the `.fls` file that the `-recorder` option writes. Build tools that read it, such as latexmk and the LaTeX Workshop extension of Visual Studio Code, therefore rebuild the document when the CFF file changes.

3.8 Checking the title page

By default, import-citation-cff checks that `\maketitle` typesets the metadata of the citation. The `\title` command should use `\cff{title}`, and the `\author` command should use `\cffauthor`, or `\cff` with a path such as `authors/1/family-names`, for every author of the citation. The check cannot see metadata that is expanded before `\maketitle`, such as an author list built with `\xdef` in the preamble, and it needs a class that keeps the title and the authors in `\@title` and `\@author`, as the standard classes do.

What the check looks for, and where, depends on the author style of the document. A style for a journal template that typesets its title block without `\maketitle` can check other fields in that block (Section 5.4). The `style` option of the package names the author style of the document, and also makes it the default style of `\cffsetauthors`:

```
\usepackage[style=authblk]{import-citation-cff}
```

The check uses this style even if `\cffsetauthors` selects another one. Without the option, the check uses the style of the last `\cffsetauthors` in the preamble, or else `standard`. Section 4 tells what each built-in style checks.

When the check does not fit the document, load the package with the `check=false` option:

```
\usepackage[check=false]{import-citation-cff}
```

3.9 Expanding metadata

The Lua command-line tool `cfflatexpand` replaces metadata commands with their values and removes the import-citation-cff package load and `\includecff`. The resulting source can be compiled without import-citation-cff or the CFF file. Install the command as described in the README, then run either

```
cfflatexpand manuscript.tex > submission.tex
cfflatexpand manuscript.tex --output submission.tex
cfflatexpand manuscript.tex -o submission.tex
```

From a source checkout, use `./cfflatexpand` instead. The command runs with `texlua`. Converting `\cffsetauthors` also requires LuaLaTeX, which runs with shell escape disabled.

`cfflatexpand` also converts `\cffsetauthors` with the keys of Section 3.5, the built-in styles of Section 4, and author style plugins that follow the rules of Section 5.5. The

`style` option of the package is its default style, as when compiling. The output still needs the class or package that provides its declarations.

For a manuscript split into several TeX files, first merge it with latexpand. Keep the merged file beside the root manuscript and keep the literal CFF filename valid there:

```
latexpand manuscript.tex > manuscript-flat.tex
cfflatexpand manuscript-flat.tex --output submission.tex
```

4 Built-in author styles

import-citation-cff comes with the following author styles. A document selects one with the `style` key of `\cffsetauthors` or the `style` option of the package (Sections 3.5 and 3.8). The other keys of `\cffsetauthors` override the defaults of the style.

4.1 `standard`

The default style collects the authors in one `\author` of the standard classes, separated by `\and`, without the affiliations. The footnotes of the contacts and the notes are typeset by `\maketitle`, like those of `\thanks`. The check looks for the title in `\title` and every author in `\author`.

4.2 `authblk`

The style declares each author with `\author` and each distinct affiliation with `\affil` of the authblk package, which lists the affiliations below the authors. The marks of an author are the letters of their affiliations and the footnotes of the contacts and the notes. Load authblk before `\cffsetauthors`. The check is that of `standard` (Section 4.1). The title page of this manual is made by

```
\usepackage{authblk}
\includecff{CITATION.cff}
\title{\cff{title}}
\cffsetauthors[style=authblk]
\hypersetup{pdfauthor={\cffauthornames}}
```

4.3 `rsc`

The style is for the article template of the Royal Society of Chemistry [3], which typesets its title block by hand in a `@twocolumnfalse` environment, without `\maketitle`. It defines two commands for the template: `\rscauthors`, the line of the authors, and `\rscfootnotes`, the footnotes that the template lists after the block.

```
\usepackage[style=rsc]{import-citation-cff}
\includecff{CITATION.cff}
\cffsetauthors[
  note = {1}{These authors contributed equally to this work.},
  note = {2}{These authors contributed equally to this work.},
]
...
 & \noindent\large{\rscauthors} \\
...
```

```
%%%FOOTNOTES%%%
\rscfootnotes
```

The authors are separated by commas, with "and" before the last. Their marks are ∗ for a contact, the letters of the affiliations, and ‡, §, and ¶ for the notes. The template keeps † for the electronic supplementary information. Authors with the same note share its symbol, and with one affiliation, no letter is needed. `\rscfootnotes` lists the contacts, then the affiliations, then the notes, each as `\footnotetext{\textit{$^{`⟨*mark*⟩`}$~`⟨*text*⟩`}}`. The default contact template is `Corresponding author. E-mail: #1{email}`.

The check looks for the title, every author, and the abstract in the `@twocolumnfalse` block, for example through `\rscauthors`, and in the whole document if there is no such block.

5 Author style plugins

An author style plugin is a definition file that registers a style for the `style` key of `\cffsetauthors` and the `style` option of the package. The style declares the authors, and also tells the check of Section 3.8 what to look for. The built-in styles of Section 4 are plugins written with the same interface.

5.1 Writing a definition file

The support files of import-citation-cff share the short prefix `cfflatex`. A style named ⟨*name*⟩ lives in the file `cfflatex-style-`⟨*name*⟩`.def`, beside the manuscript or in a TeX tree, and registers itself with `\cffdeclareauthorstyle`. For example, the following style uses the `\author` and `\affil` commands of authblk with numeric affiliation marks:

```
% cfflatex-style-authblk-arabic.def
\cffdeclareauthorstyle{authblk-arabic}{
  requires = {authblk},
  affiliation-mark = arabic,
  author = {\author[{{}#1}]{#2}},
  affiliation = {\affil[#1]{#2}},
}
```

The manuscript selects it by name:

```
\usepackage{authblk}
\usepackage{import-citation-cff}
\includecff{CITATION.cff}
\cffsetauthors[style=authblk-arabic]
```

`\cffsetauthors`, or the `style` option of the package at `\begin{document}`, reads the definition file the first time the style is used, and the file must register the requested name. Names are case-sensitive, start with an ASCII letter or digit, and contain only ASCII letters, digits, hyphens, and underscores. Registration is global, and registering a name twice is an error. Define helper commands that are used outside the file globally. Definition files are read with normal document catcodes. Put `\ExplSyntaxOn` and `\ExplSyntaxOff` around expl3 code.

5.2 Plugin keys

A style definition accepts the keys of `\cffsetauthors`, except `style`, as its defaults. Each call of `\cffsetauthors` resets every key, then applies the style definition, then the user's options. User options therefore override the style even when they appear before `style` in the option list.

The following keys are only useful in a style definition.

`requires` a comma-separated list of packages that must already be loaded. The plugin does not load them.

`validate` code that checks the document, such as the class or the fields the style needs.

`begin`, `end` code run before and after the declarations. For example, the `standard` style collects the names and declares one `\author` in its `end` code.

`prepare-author` code run before each author is declared, with `#1` equal to the index of the author. Use the public accessors of Section 6 to inspect that author.

`author-code`, `affiliation-code` code run in place of the `author` and `affiliation` templates, with the same arguments.

`render` code that replaces the default order of declarations. An empty value selects the default.

`check-fields`, `check-region` what the check looks for (Section 5.4). These keys are only allowed in a style definition.

The `author` and `affiliation` templates are declarations: `\cffsetauthors` passes each of them to `\cffemit` itself. The `author-code` and `affiliation-code` keys are code instead: they can prepare or accumulate, and must pass each final declaration to `\cffemit`. An empty affiliation template or code gives no affiliations and no affiliation marks.

5.3 Rendering order

`\cffsetauthors` runs a style in this order:

1. the `requires` check and the `validate` code;
2. the collection of the names for `\cffauthornames`;
3. the `begin` code;
4. the renderer, which by default calls `\cffrenderauthor` for every author and then `\cffrenderaffiliation` for every distinct affiliation;
5. the `end` code.

`\cffauthornames` is available from the `begin` code on, and does not depend on the order of the renderer. `\cffrenderauthor` runs the `prepare-author` code before the author template. A `render` value can use the loops of Section 6 to change the order. For example, this renderer declares each author followed by the affiliations of that author, repeating shared affiliations:

```
render = {
  \cffforeachauthor{
    \cffrenderauthor{\cffauthorindex}
    \cffforeachauthoraffiliation{\cffauthorindex}{
      \cffrenderaffiliation{\cffconvertaffiliationindex
        {\cffauthorindex}{\cffauthoraffiliationindex}}}
  }
},
```

5.4 Checking the title page

When a style is the author style of the document (Section 3.8), the check reads two keys of its definition.

check-fields the fields that the title page must use, given as paths of `\cff`, such as `title` and `abstract`, where `authors` stands for every author. The default is `{title, authors}`. A field that the citation does not have is not checked.

check-region the regions of the document where the fields must be used, one or more of

- **maketitle** `\maketitle`, where `\title` must use the title and `\author` every author. This is the default.
- **hooks={**⟨*begin*⟩**}{**⟨*end*⟩**}** from the LaTeX hook ⟨*begin*⟩ to the hook ⟨*end*⟩.
- **document** the whole document, as a fallback.

The first region that ends is checked, and no region is checked after it. If no other region has ended, `document` is checked at the end of the document, and the log says so. A use counts when `\cff`, `\cffauthor`, or `\cffauthornames` is typeset inside the region, also through macros of the document.

For example, the template of the Royal Society of Chemistry typesets its title block in a `@twocolumnfalse` environment. Every environment has hooks, so the `rsc` style (Section 4.3) checks this block with

```
check-fields = {title, authors, abstract},
check-region = {
  hooks = {env/@twocolumnfalse/begin}{env/@twocolumnfalse/end},
  document,
},
```

The check does not run during export, so these keys do not affect `cfflatexpand`.

5.5 Supporting `cfflatexpand`

`cfflatexpand` converts `\cffsetauthors` by running the style, not by reading it. It

1. starts a separate LuaLaTeX run of an empty `article` document that loads import-citation-cff and the metadata, but none of the manuscript's preamble;
2. calls `\cffsetauthors` there with the options of the manuscript, and the `style` option of the package as the default, loading the definition file from the directory of the manuscript or the TeX search path, and skipping `requires` and `validate`;

3. records the argument of every `\cffemit` as it is, without executing it;
4. replaces `\cffsetauthors` in the output with the recorded declarations, in which the metadata commands become their values and the footnotes of contacts and notes use a self-contained definition.

A style therefore works with `cfflatexpand` if it follows these rules.

- Register the style in a definition file. `\cffdeclareauthorstyle` in the manuscript works when compiling, but `cfflatexpand` reports an error.
- Pass every final declaration to `\cffemit`. A declaration that the code executes directly, such as `\author` in `author-code`, appears when compiling but is missing from the output.
- Make the preparation code self-contained. The `begin`, `end`, `prepare-author`, `author-code`, `affiliation-code`, and `render` code also runs in the isolated document, where the class and the packages of the manuscript, including those in `requires`, are not loaded.
- Use only commands of the final class and its packages in the declarations. The output is compiled without import-citation-cff and without the definition file, so helper commands of the style are undefined there. A declaration can define a command for the document with `\newcommand`, as the `rsc` style does.
- Keep the marks and the names unexpanded. They contain the footnotes and the metadata commands, which must run when the title page is typeset. To accumulate declarations, append the arguments with `\tl_put_right:Nn` and insert the result with `\exp_not:V`, as `cfflatex-style-standard.def` does, rather than expanding the whole declaration with `\edef`.

6 Reference

The package has two options (Section 3.8).

check=⟨*boolean*⟩ checks the title page. The default is `true`.

style=⟨*name*⟩ sets the author style of the document, which the check uses and `\cffsetauthors` selects by default.

`\includecff` `\includecff [`⟨*options*⟩`] {`⟨*file*⟩`}`

Reads the citation metadata in ⟨*file*⟩. The command can be used in the preamble or in the document. Each call replaces the metadata read by the previous one.

source=⟨*key*⟩ reads the mapping under the top-level ⟨*key*⟩ instead. Without this option, the citation is the top level of the file.

`\cff` ⋆ `\cff {`⟨*path*⟩`}`
`\cffcount` ⋆ `\cffcount {`⟨*path*⟩`}`

Expand to the value at ⟨*path*⟩ in the citation, and to the length of the list at ⟨*path*⟩. ⟨*path*⟩ lists the keys and the indices, counting from 1, that lead to the value, separated by slashes, such as `authors/1/family-names`. A missing path gives nothing for `\cff` and 0 for `\cffcount`. A list or a mapping for `\cff`, and a value or a mapping for `\cffcount`, is an error. `\cff{authors/`⟨*index*⟩`/`⟨*field*⟩`}` is `\cffauthor{`⟨*index*⟩`}{`⟨*field*⟩`}` without the check of the index. Section 3.2 shows examples.

`\cffsetauthors` `\cffsetauthors [`⟨*keys*⟩`]`
`\cffauthornames` `\cffauthornames`

`\cffsetauthors` declares the authors of the citation for `\maketitle`, as Section 3.5 describes. Call this function once after `\includecff`. `\cffauthornames` expands to the names of the authors, separated by commas and made by the `name` template. The ⟨*keys*⟩ are

`style=`⟨*name*⟩ selects a registered author style, loading its definition file if needed. The default is the `style` option of the package, or else `standard`. Section 4 lists the built-in styles, and Section 5 explains how to write others.

`name=`⟨*template*⟩ makes the name of an author. `#1` takes a field of the author. The default is `#1{given-names}~#1{family-names}`.

`affiliation-mark=`⟨*style*⟩ numbers the affiliations with `alph` (the default), `Alph`, `arabic`, `roman`, or `Roman`.

`separator=`⟨*text*⟩ separates the marks of an author. The default is a comma.

`contact=`⟨*template*⟩ is the footnote of an author who is a contact. `#1` takes a field of the contact. The default is `E-mail: #1{email}`, and an empty template gives no footnotes.

`note={`⟨*index*⟩`}{`⟨*text*⟩`}` adds a footnote with ⟨*text*⟩ to the ⟨*index*⟩-th author, such as a current affiliation. ⟨*index*⟩ is an integer expression, such as `\cffcount{authors}` for the last author. The key can be given several times. The footnotes follow those of the contacts, in order.

`author=`⟨*template*⟩ declares an author, with the marks `#1` and the name `#2`.

`affiliation=`⟨*template*⟩ declares an affiliation, with the mark `#1` and the name `#2`. An empty template gives no affiliations.

`\cffdeclareauthorstyle` `\cffdeclareauthorstyle {`⟨*name*⟩`} {`⟨*keys*⟩`}`

Registers an author style globally. See Section 5 for naming rules, definition files, templates, lifecycle hooks, and the keys of the check.

`\cffemit` `\cffemit {⟨declaration⟩}`
`\cffrenderauthor` `\cffrenderauthor {⟨index⟩}`
`\cffrenderaffiliation` `\cffrenderaffiliation {⟨index⟩}`

Plugin code sends a final declaration to `\cffemit`, which executes it during normal compilation and collects it during export. In a custom renderer, `\cffrenderauthor` and `\cffrenderaffiliation` apply the current templates to an author or a distinct affiliation. Their indices are integer expressions and use the same numbering as the public field accessors.

`\cffauthor` ⋆ `\cffauthor {⟨index⟩} {⟨field⟩}`

Expands to the ⟨*field*⟩ of the ⟨*index*⟩-th author, counting from 1, and reports an error if there is no such author. ⟨*field*⟩ is a key of the author entry in the CFF file, such as `given-names` or `family-names`. A field that the author does not have expands to nothing. `\cffcount{authors}` is the number of authors.

`\cffforeachauthor` `\cffforeachauthor {⟨code⟩}`
`\cffauthorindex` `\cffauthorindex`

`\cffforeachauthor` runs ⟨*code*⟩ once for each author of the citation, in order, and `\cffauthorindex` expands to the index of the current author, counting from 1.

`\cffauthoraffiliationcount` ⋆ `\cffauthoraffiliationcount {⟨index⟩}`
`\cffauthoraffiliation` ⋆ `\cffauthoraffiliation {⟨index⟩} {⟨k⟩}`
`\cffconvertaffiliationindex` ⋆ `\cffconvertaffiliationindex {⟨index⟩} {⟨k⟩}`

Expand to the number of affiliations of the ⟨*index*⟩-th author, which is 0 if the author has none, to the ⟨*k*⟩-th of them, and to its number among the distinct affiliations of the citation, which `\cffaffiliation` and `\cffforeachaffiliation` use.

`\cffforeachauthoraffiliation` `\cffforeachauthoraffiliation {⟨index⟩} {⟨code⟩}`
`\cffauthoraffiliationindex` `\cffauthoraffiliationindex`

`\cffforeachauthoraffiliation` runs ⟨*code*⟩ once for each affiliation of the ⟨*index*⟩-th author, in order, and `\cffauthoraffiliationindex` expands to the index of the current affiliation among those of the author, counting from 1.

`\cffaffiliationcount` ⋆ `\cffaffiliationcount`
`\cffaffiliation` ⋆ `\cffaffiliation {⟨n⟩}`

Expand to the number of distinct affiliations of the citation, and to the ⟨*n*⟩-th of them.

`\cffforeachaffiliation` `\cffforeachaffiliation {⟨code⟩}`
`\cffaffiliationindex` `\cffaffiliationindex`

`\cffforeachaffiliation` runs ⟨*code*⟩ once for each distinct affiliation of the citation, in order, and `\cffaffiliationindex` expands to the number of the current affiliation.

`\cffcontact` ⋆ `\cffcontact {⟨index⟩} {⟨field⟩}`

Expands to the ⟨*field*⟩ of the ⟨*index*⟩-th contact, counting from 1. As for `\cffauthor`, ⟨*field*⟩ is a key of the contact entry, such as `email`, and a field that the contact does not have expands to nothing. `\cffcount{contact}` is the number of contacts, which is 0 if there are none.

`\cffforeachcontact` `\cffforeachcontact {⟨code⟩}`
`\cffcontactindex` `\cffcontactindex`

`\cffforeachcontact` runs ⟨*code*⟩ once for each contact of the citation, in order, and `\cffcontactindex` expands to the index of the current contact, counting from 1.

7 Errors and warnings

The following errors stop compilation.

CFF file '⟨*file*⟩' was not found. ⟨*file*⟩ is not in the current directory or the TEX search path.

pdfLaTeX requires unrestricted shell escape ... Use LuaLATEX, or run pdfLATEX with `--shell-escape`.

Cannot read CFF file: ⟨*reason*⟩. The file cannot be used. For example, it has no top-level key given by `source`, or that key does not hold a mapping, or the selected citation has incorrect structure, or an affiliation breaks the rules of Section 3.4.

Install cfflatex.lua alongside import-citation-cff.sty. import-citation-cff is incompletely installed.

The following errors are reported while using the commands of Section 6.

Call `\includecff` first. No CFF file has been read yet.

style=⟨*name*⟩ needs the ⟨*package*⟩ package. Load ⟨*package*⟩, which the `requires` key of the style lists, before `\cffsetauthors`.

Invalid author style name '⟨*name*⟩'. The name breaks the rules of Section 5.

Author style '⟨*name*⟩' is not registered ... The definition file of the style was not found.

Author style '⟨*name*⟩' was not registered by its file. The definition file registers another name.

Author style '⟨*name*⟩' is already registered. Each name can be registered once.

The key '⟨*key*⟩' is only allowed in an author style definition. The check keys are given to `\cffsetauthors`.

Invalid check region '⟨*region*⟩'. The `check-region` key of the author style of the document lists an unknown region, or hooks without exactly two arguments.

Author index '⟨*index*⟩' is out of range. The citation has fewer than ⟨*index*⟩ authors.

Affiliation '⟨*k*⟩' is out of range. The author or the citation has fewer than ⟨*k*⟩ affiliations.

Contact '⟨*index*⟩' is out of range. The citation has fewer than ⟨*index*⟩ contacts.

Not inside `\cffforeachauthor`. `\cffauthorindex` is used outside the loop.

Not inside `\cffforeachcontact`. `\cffcontactindex` is used outside the loop.

Not a value: ⟨*path*⟩. `\cff` is given the path of a list or a mapping.

Not a list: ⟨*path*⟩. `\cffcount` is given the path of a value or a mapping.

Not inside an affiliation loop. `\cffaffiliationindex` is used outside the loops over affiliations.

Not in an author affiliation loop. The index of an affiliation of an author is used outside the loop over the affiliations of the author.

The following warnings are issued by the check of Section 3.8, after the region it checks, such as `\maketitle`.

`\title` does not use the title of the citation. The title page does not show the title of the citation.

`\author` does not use author(s) ⟨*indices*⟩ of the citation. The title page does not show these authors.

The title page does not use '⟨*field*⟩' of the citation. The region does not show this field.

The title page does not use author(s) ... The region does not show these authors.

8 License

import-citation-cff is distributed under the MIT License. Its source code is at https://github.com/JSS95/cfflatex, where bugs can be reported.

References

[1] Arfon M. Smith, Daniel S. Katz, Kyle E. Niemeyer, and FORCE11 Software Citation Working Group. Software citation principles. *PeerJ Computer Science*, 2:e86, 2016. https://doi.org/10.7717/peerj-cs.86.

[2] Stephan Druskat, Jurriaan H. Spaaks, Neil Chue Hong, Robert Haines, James Baker, Spencer Bliven, Egon Willighagen, David Pérez-Suárez, and Olexandr Konovalov. Citation File Format. Version 1.2.0, 2021. https://doi.org/10.5281/zenodo.5171937.

[3] Royal Society of Chemistry. Royal Society of Chemistry article template. Overleaf, 2019. https://www.overleaf.com/latex/templates/royal-society-of-chemistry-article-template/vffpnxpmxdfz.