`bibcop`: LaTeX Package for Style Checking of `.bib` Files*

Yegor Bugayenko
yegor256@gmail.com

2026-10-16, 0.0.37

NB! The TeX processor must be invoked with the `--shell-escape` option, and Perl must be installed on the system. This package does not operate on Windows.

1 Introduction

This package scans a `.bib` file for stylistic errors and emits warning messages whenever issues are detected (the package must be included before all other bibliography-related packages):

```
\documentclass{article}
\usepackage{bibcop}
\begin{document}
\bibliographystyle{plain}
\bibliography{main}
\end{document}
```

Warnings, if any, appear in the TeX log. Once the issues in the `main.bib` file have been resolved, the warnings cease.

Bibcop is largely indifferent to the formatting of the `.bib` file. It does not emit warnings when a comma is omitted after the final tag of a BibTeX entry, nor when the tags and their equals symbols are misaligned. Bibcop's attention is reserved for the content of the tags and the inconsistencies that may arise within them.

Whenever the `.sty` file is used directly (without installation into the TeX tree), the `bibcop.pl` file must accompany it—it is the Perl script that performs the actual checking of `.bib` files. The `.sty` serves merely as a thin wrapper around it.

The `\usepackage{bibcop}` declaration must immediately follow `\usepackage{biblatex}` (whenever BibLaTeX is in use); otherwise no warnings will be issued by `bibcop`.

*The sources are in GitHub at yegor256/bibcop

2 Package Options

The behaviour of the package may be configured by means of several package options:

`verbose` The `verbose` package option prints all debugging messages to the TEX log:

```
\usepackage[verbose]{bibcop}
```

`script` The package relies upon the `bibcop.pl` file, a Perl script that performs all of the work. This file is expected to reside either in the current directory or within the `texmf-dist/scripts/bibcop/` directory. By means of the `script` option, the location of the script may be specified explicitly (the option is not recommended unless a particular need arises):

```
\usepackage[script=my-perl-script.pl]{bibcop}
```

`no*` Specific rules may be suppressed by means of one of the `no*` package options:

```
\usepackage[nodoi,nowraps]{bibcop}
```

The following options are available:

- `nocaps` allows arbitrary capitalization in titles;
- `nowraps` allows titles, booktitles, and journals to have no double curly braces;
- `nodoi` allows the absence of the `doi` tag in all entries;
- `noinproc` allows the `booktitle` tag in `@inproceedings` entries to not start with "Proceedings of the";
- `noorg` allows mentioning of ACM/IEEE in the `booktitle` tag;
- `notags` allows any tags and allows missing important tags.

3 The Rules

The following constitutes a substantially complete enumeration of the rules enforced on a `.bib` file:

`types` Only the `@article`, `@book`, `@inproceedings`, `@phdthesis`, `@incollection`, and `@misc` types of BIBTEX entries are admitted. All other types, such as `@manual`, `@inbook`, and many others, are forbidden. The aforementioned six suffice for every conceivable purpose.

`tags` Each type of BIBTEX entry has a tightly constrained list of admissible tags. Tags absent from the list are forbidden.

`doi` Every BIBTEX entry must contain a `doi` tag, namely the unique Digital Object Identifier of the cited material. To forestall ambiguity, it is sound practice to record the DOI in every case. The doi package is likewise recommended, as it converts all "`doi`" fields into hyperlinks.

`caps` Within the `title`, `booktitle`, `journal`, `publisher`, and `organization` tags, every major word must be capitalised, as recommended by the APA:

```
title = {A Preliminary Architecture for a Basic Data-Flow Processor}
```

Here, the leading "A" is capitalised because it opens the title. The word "for" and the article "a" are minor words and therefore appear in lower case. Both components of the composite word "Data-Flow" are capitalised. This rule may occasionally be relaxed where a custom capitalisation is preserved as imposed by the author of the paper. In such cases, the words bearing custom capitalisation must be enclosed in curly brackets, for example:

```
title = {Structured Programming {with} Go {To} Statements}
```

An entire value may be protected in the same manner, by one extra pair of curly brackets around it (the pair that these three tags demand anyway does not count):

```
title = {{{lowercase title of a paper}}}
```

This rule may be disabled by means of the `nocaps` package option.

`author` The `author` field must contain a list of authors separated by "`and`". Each author must comprise two components separated by a comma: the first denotes the surname of the author, the second a list of given names separated by spaces, for example:

```
author = {Knuth, Donald E. and Duane, Bibby}
```

Where the list of authors is exceedingly long, "`and others`" may be employed:

```
author = {Knuth, Donald E. and others}
```

Whenever given names are abbreviated to a single letter, a trailing full stop is required. An author name with bespoke formatting may be enclosed in curly brackets, whereupon Bibcop disregards it:

```
author = {{Some author} and {I}}
```

`shorts` Abbreviation of words is permitted only within the `author` tag; the following, for instance, is impermissible:

```
journal = {J. Log. Compute}
```

It must instead be replaced by the following:

```
journal = {Journal of Logic and Computation}
```

`brackets` The `title`, `booktitle`, and `journal` fields must be enclosed in double brackets, for example:

```
title = {{A Survey of Symbolic Execution Techniques}}
```

This is necessary in order to prevent the down-casing of capitalised words, which certain bibliography styles otherwise perform.

`year` The year may not appear within the title of a conference or a journal; the following, for instance, is impermissible:

```
booktitle = {{1994 IEEE International Conference on Computer Languages}},
```

The year is to be recorded solely within the `year` tag, and nowhere else. Within the `year` tag, only numerals are admissible:

```
year = {1994},
```

`month` The `month` field may contain only a numeral:

```
month = {12},
```

`volume` The `volume` field may contain only a numeral:

```
volume = {32},
```

`number` The `number` field may contain only a numeral:

```
number = {132},
```

`pages` The `pages` field may contain either a single numeral or two numerals separated by a double dash:

```
pages = {145--163},
```

`proceedings` The `booktitle` within an `@inproceedings` entry must invariably begin with "`Proceedings of the`", as illustrated below:

```
booktitle = {{Proceedings of the International
  Conference on Computer Languages}},
```

This rule may be disabled by means of the `noinproc` package option.

`arXiv` Whenever `archivePrefix` is present, `eprint` and `primaryClass` must likewise be present and must conform to the formatting principles of arXiv identifiers:

```
@misc{bugayenko2021,
  archivePrefix = {arXiv},
  eprint = {2111.13384},
  primaryClass = {cs.PL},
}
```

`typography` All tags within each BibTeX entry are inspected for compliance with elementary typographic conventions:

- No space is admissible before a comma, semicolon, colon, full stop, question mark, or exclamation mark;
- Text may not terminate in a full stop, comma, semicolon, colon, or dash;
- A triple dash must be flanked on both sides by spaces.

`Unicode` Non-ASCII characters are prohibited throughout the `.bib` file, unless curled brackets protect them:

```
@book{proust1913,
  author = {{Proust, Marcel}},
  title = {{{À la Recherche du Temps Perdu}}},
  publisher = {{Éditions Grasset}},
}
```

The brackets that delimit the value of a tag protect nothing, which is why one extra pair is required around the text itself (and yet one more around a `title`, since it demands its own pair anyway). This is how a bibliography in Cyrillic, Greek, Arabic, or Chinese may be kept, even though transliteration into Latin remains the safer choice.

4 Implementation

First, we include a few packages. We need iexec for executing Perl scripts:

```
\RequirePackage{iexec}
```

Then, we process package options:

```
\RequirePackage{pgfopts}
\pgfkeys{
  /bibcop/.cd,
  notags/.store in=\bibcop@notags,
  noorg/.store in=\bibcop@noorg,
  noinproc/.store in=\bibcop@noinproc,
  nocaps/.store in=\bibcop@nocaps,
  nodoi/.store in=\bibcop@nodoi,
  nowraps/.store in=\bibcop@nowraps,
  verbose/.store in=\bibcop@verbose,
  script/.store in = \bibcop@script,
}
\ProcessPgfPackageOptions{/bibcop}
```

`bibcop.pl` Then, we find the Perl script:

```
\makeatletter
\ifdefined\bibcop@script\else
  \IfFileExists{bibcop.pl}
    {\gdef\bibcop@script{perl ./bibcop.pl}}
    {\gdef\bibcop@script{bibcop}}
\fi
\message{bibcop: The Perl script is at '\bibcop@script'^^J}%
\makeatother
```

`\bibcop@exec` Then, we define a supplementary command to execute the Perl script:

```
\RequirePackage{shellesc}
\makeatletter
\newcommand\bibcop@exec[1]{
  \iexec[maybe]{\bibcop@script\space
    \ifdefined\bibcop@verbose--verbose\fi\space
    \ifdefined\bibcop@notags--no:tags\fi\space
    \ifdefined\bibcop@noorg--no:org\fi\space
    \ifdefined\bibcop@noinproc--no:inproc\fi\space
    \ifdefined\bibcop@nodoi--no:doi\fi\space
    \ifdefined\bibcop@nocaps--no:caps\fi\space
    \ifdefined\bibcop@nowraps--no:wraps\fi\space
    --latex '#1'}%
  \message{bibcop: style checking finished for #1^^J}%
}
\makeatother
```

`\bibliography` Then, we re-define the `\bibliography` command:

```
\makeatletter
\ifdefined\bibliography
  \let\bibcop@oldbibliography\bibliography
  \renewcommand\bibliography[1]{%
    \bibcop@exec{#1.bib}%
    \bibcop@oldbibliography{#1}%
```

```
  }
\fi
\makeatother
```

`\addbibresource` Then, we re-define the `\addbibresource` command:

```
\makeatletter
\ifdefined\addbibresource
  \let\bibcop@oldaddbibresource\addbibresource
  \renewcommand\addbibresource[1]{%
    \bibcop@exec{#1}%
    \bibcop@oldaddbibresource{#1}%
  }
\fi
\makeatother

\endinput
```

Change History

0.0.1

General: First draft. 5

0.0.12

General: A few package options introduced to give users an ability to disable certain style rules: `nocaps`, `nowraps`, `notags`, `noorg`, `noinproc`, and `nodoi`. Also, a command line script gets a new set of options, which start from `-no:`, for example `-no:caps`. 5

0.0.16

`\bibcop@exec`: When `-shell-escape` is not set, there is no validation performed. . . . 5

0.0.2

General: Documentation extended, more rules added. 5

0.0.4

General: Extra checks for the typography, together with more extensive Perl testing. 5

Package options introduced, the `verbose` option enables detailed logging inside the TEX log. . . . 5

The `-verbose` option introduced, to enable debugging information only on demand. . . 5

0.0.6

General: The `script` package option introduced, to enable explicit configuration of the location of the `bibcop.pl` Perl script. 5

Index

Numbers written in italic refer to the page where the corresponding entry is described; numbers underlined refer to the code line of the definition; numbers in roman refer to the code lines where the entry is used.

A

\addbibresource . . . 47
\arXiv 4
\author 3

B

\bibcop.pl 15
\bibcop@exec . 23, 42, 51
\bibcop@nocaps . . . 8, 32
\bibcop@nodoi 9, 31
\bibcop@noinproc . 7, 30
\bibcop@noorg 6, 29
\bibcop@notags . . . 5, 28
\bibcop@nowraps . 10, 33
\bibcop@oldaddbibresource 49, 52
\bibcop@oldbibliography 40, 43
\bibcop@script . 12, 16, 18, 19, 21, 26
\bibcop@verbose . 11, 27
\bibliography 38
\brackets 3

C

\caps 2

D

\doi 2

E

\endinput 56

G

\gdef 18, 19

I

\iexec 26
\ifdefined 16, 27, 28, 29, 30, 31, 32, 33, 39, 48
\IfFileExists 17

M

\makeatletter 15, 24, 38, 47
\makeatother 22, 37, 46, 55
\message 21, 35
\month 4

N

\newcommand 25
\no* 2
\number 4

P

\pages 4
\pgfkeys 3
\proceedings 4
\ProcessPgfPackageOptions 14

R

\renewcommand . . . 41, 50
\RequirePackage 1, 2, 23

S

\script 2
\shorts 3
\space 26, 27, 28, 29, 30, 31, 32, 33

T

\tags 2
\types 2
\typography 4

U

\Unicode 4

V

\verbose 2
\volume 4

Y

\year 3