



# bibcop: L<sup>A</sup>T<sub>E</sub>X Package for Style Checking of .bib Files\*

Yegor Bugayenko  
yegor256@gmail.com

2024-03-15, 0.0.20

**NB!** You must run T<sub>E</sub>X processor with `--shell-escape` option and you must have [Perl](#) installed. This package doesn't work on Windows.

## 1 Introduction

This package scans a .bib file for style errors and emits warning messages if any issues are found (the package must be included before all other bibliography related packages):

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

Some warnings may be printed in the T<sub>E</sub>X log. Once the issues in the main.bib file are fixed, the warnings disappear.

Bibcop doesn't pay much attention the formatting of the .bib file. It doesn't emit warnings if a comma is missed after the last tag in a BibT<sub>E</sub>X entry or the tags and their equation symbols are not aligned. Instead, Bibcop is focused on the content of the tags and their possible inconsistencies.

If the .sty file is used directly (without installing it into the T<sub>E</sub>X tree), the bibcop.pl file must also be placed next to it – it is the Perl script that does all the work of checking your .bib files. The .sty is just a simple wrapper around it.

The `\usepackage{bibcop}` must stay right after `\usepackage{biblatex}` (if BibT<sub>E</sub>X is used), otherwise there won't be any warnings from bibcop.

---

\*The sources are in GitHub at [yegor256/bibcop](https://github.com/yegor256/bibcop)

## 2 Package Options

It's possible to configure the behavior of the package with the help of a few package options:

`verbose` The `verbose` package option prints all debugging messages to the  $\TeX$  log:

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

`script` The package depends on the `bibcop.pl` file, which is a Perl script that does all the work. This file is supposed to be located either in the current directory or in the `texmf-dist/scripts/bibcop/` directory. Using the `script` option the location of the script may be explicitly defined (it is not recommended to use this option unless there is a special demand for it):

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

`no*` It's possible to suppress certain rules, by using one of the `no*` package options:

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

The following options are available:

- `nocaps` allows arbitrary capitalization in titles;
- `nowraps` allows titles 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 allow to miss important tags.

## 3 The Rules

This is a more or less complete list of rules enforced on a `.bib` file:

`types` Only `@article`, `@book`, `@inproceedings`, and `@misc` types of  $\text{BIB}\TeX$  entries are allowed. Everything else, like `@manual`, `@phdthesis`, and many others are simply prohibited. The mentioned four should be enough for everything.

`tags` There are pretty limited lists of allowed tags for each type of  $\text{BIB}\TeX$  entry. The tags that are not in the list are prohibited to use.

`doi` Every  $\text{BIB}\TeX$  entry must have the `doi` tag, which is a unique [Digital Object Identifier](#) of the material that you reference. It seems to be a good practice, in order to avoid ambiguity, to always mention the DOI. It also recommended to use the `doi` package, in order to make all "doi" fields turned into hyper links.

`caps` In the `title`, `booktitle`, and `journal` tags, all major words must be capitalized, as it is [recommended](#) by APA:

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

Here, the leading "A" is capital because it opens the title. The word "for" and the article "a" are minor words, that's why they are in lower case. Both parts of the composite word "Data-Flow" are capitalized. Sometimes this rule may need to be violated, when there is custom capitalization, as it is done by the author of the paper. In order to do this, the words with custom capitalization must be wrapped in curled brackets, for example:

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

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

**author** The `author` must contain a list of authors separated by “and”. Each author must have two parts separated by a comma. The first part is the last name of the author, the second part is a list of their first names separated by a space, for example:

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

When the list of authors is too long, it’s possible to say “and others”:

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

When first names are shortened to a single letter, it has to have a trailing dot. A specially pronounced author may be wrapped it in curled brackets, to make Bibcop ignore it:

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

**shorts** It is not allowed to shorten any words aside from the `author` tag, for example this is illegal:

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

This must be replaced with the following:

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

**brackets** The `title`, `booktitle`, and `journal` must be wrapped in double brackets, for example:

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

This is necessary in order to prevent down-casing of capitalized words, which is done by some bibliography styles.

**year** It is not allowed to mention the year inside the title of a conference or a journal, for example, this would be illegal:

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

The year should only be mentioned in the `year` tag, nowhere else. In the `year` tag only numbers are allowed:

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

**month** The `month` may contain only a number:

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

**volume** The `volume` may contain only a number:

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

**number** The `number` may contain only a number:

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

**pages** The `pages` may contain either a number or two numbers separated by a double dash:

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

`proceedings` The `booktitle` in the `@inproceedings` entry must always start with “Proceedings of the”, as in this example:

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

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

`arXiv` If the `archivePrefix` is present, the `eprint` and the `primaryClass` must also be present and must adhere to the formatting principles of [arXiv identifiers](#):

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

`typography` All tags in each `BIBTEX` entry are checked for obeying the basic typography rules:

- No spaces are allowed in front of a comma, a semi-colon, a colon, a dot, a question mark, and an exclamation mark;
- A text may not end with a dot, a comma, a semi-colon, a colon, or a dash;
- A triple dash must be surrounded by spaces.

`Unicode` Any non-ASCII characters are prohibited in the entire `.bib` file.

## 4 Implementation

First, we include a few packages. We need [iexec](#) for executing Perl scripts:

```
1 \RequirePackage{iexec}
   Then, we process package options:
2 \RequirePackage{pgfopts}
3 \pgfkeys{
4   /bibcop/.cd,
5   notags/.store in=\bibcop@notags,
6   noorg/.store in=\bibcop@noorg,
7   noinproc/.store in=\bibcop@noinproc,
8   nocaps/.store in=\bibcop@nocaps,
9   nodoi/.store in=\bibcop@nodoi,
10  nowraps/.store in=\bibcop@nowraps,
11  verbose/.store in=\bibcop@verbose,
12  script/.store in = \bibcop@script,
13 }
14 \ProcessPgfPackageOptions{/bibcop}
```

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

```
15 \makeatletter
16 \ifdefined\bibcop@script\else
17   \IfFileExists{bibcop.pl}
18     {\gdef\bibcop@script{perl ./bibcop.pl}}
19     {\gdef\bibcop@script{bibcop}}
20 \fi
21 \message{bibcop: The Perl script is at '\bibcop@script'^^J}%
22 \makeatother
```

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

```
23 \RequirePackage{shellesc}
24 \makeatletter
25 \newcommand\bibcop@exec[1]{
26   \ifdefined\bibcop@script\space
27   \ifdefined\bibcop@verbose--verbose\fi\space
28   \ifdefined\bibcop@notags--no:tags\fi\space
29   \ifdefined\bibcop@noinproc--no:org\fi\space
30   \ifdefined\bibcop@noinproc--no:inproc\fi\space
31   \ifdefined\bibcop@nodoi--no:doi\fi\space
32   \ifdefined\bibcop@nocaps--no:caps\fi\space
33   \ifdefined\bibcop@nowraps--no:wraps\fi\space
34   --latex '#1'%
35   \message\bibcop: style checking finished for #1^^J}%
36 }
37 \makeatother
38
39 % \begin{macro}{\bibliography}
40 % Then, we re-define the |\bibliography| command:
41 %   \begin{macrocode}
42 \makeatletter
43 \ifdefined\bibliography
44   \let\bibcop@oldbibliography\bibliography
45   \renewcommand\bibliography[1]{%
46     \bibcop@exec{#1.bib}%
47     \bibcop@oldbibliography{#1}%
48   }
49 \fi
50 \makeatother
```

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

```
51 \makeatletter
52 \ifdefined\addbibresource
53   \let\bibcop@oldaddbibresource\addbibresource
54   \renewcommand\addbibresource[1]{%
55     \bibcop@exec{#1}%
56     \bibcop@oldaddbibresource{#1}%
57   }
58 \fi
59 \makeatother

60 \endinput
```

## Change History

0.0.1		more rules added. . . . .	4
	General: First draft. . . . .		4
0.0.12		0.0.4	
	General: A few package options introduced to give users an ability to disable certain style rules: <code>nocaps</code> , <code>nowraps</code> , <code>notags</code> , <code>noorg</code> , <code>noinproc</code> , and <code>nodoi</code> . Also, a command line scripts gets a new set of options, which start from <code>-no:</code> , for example <code>-no:caps</code> . . . . .		4
0.0.16		General: Extra checks for the typography, together with more extensive Perl testing. . . . .	4
	<code>\bibcop@exec</code> : When <code>-shell-escape</code> is not set, there is no validation performed. . . . .	Package options introduced, the <code>verbose</code> option enables detailed logging inside the $\TeX$ log. . . . .	4
0.0.2		The <code>-verbose</code> option introduced, to enable debugging information only on demand. . . . .	4
	General: Documentation extended,	0.0.6	
		General: The <code>script</code> package option introduced, to enable explicit configuration of the location of the <code>bibcop.pl</code> Perl script. . . . .	4

## 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.

<b>A</b>	<code>\bibcop@verbose</code> .. 11, 27	<code>\makeatother</code> 22, 37, 50, 59
<code>\addbibresource</code> .... <u>51</u>	<code>\bibliography</code> .....	<code>\message</code> ..... 21, 35
	... 39, 40, 43, 44, 45	
<b>B</b>		<b>N</b>
<code>\begin</code> ..... 39, 41	<b>E</b>	<code>\newcommand</code> ..... 25
<code>\bibcop.pl</code> ..... <u>15</u>	<code>\endinput</code> ..... 60	
<code>\bibcop@exec</code> ..... <u>23</u> , 55	<b>G</b>	<b>P</b>
<code>\bibcop@nocaps</code> .... 8, 32	<code>\gdef</code> ..... 18, 19	<code>\pgfkeys</code> ..... 3
<code>\bibcop@nodoi</code> ..... 9, 31	<b>I</b>	<code>\ProcessPgfPackageOptions</code>
<code>\bibcop@noinproc</code> 7, 29, 30	<code>\iexec</code> ..... 26	..... 14
<code>\bibcop@noorg</code> ..... 6	<code>\ifdefined</code> 16, 27, 28, 29,	<b>R</b>
<code>\bibcop@notags</code> .... 5, 28	30, 31, 32, 33, 43, 52	<code>\renewcommand</code> .... 45, 54
<code>\bibcop@nowraps</code> .. 10, 33	<code>\IfFileExists</code> ..... 17	<code>\RequirePackage</code> . 1, 2, 23
<code>\bibcop@oldaddbibresource</code>		
..... 53, 56	<b>M</b>	<b>S</b>
<code>\bibcop@oldbibliography</code>	<code>\makeatletter</code> .....	<code>\space</code> ..... 26, 27,
..... 44, 47	..... 15, 24, 42, 51	28, 29, 30, 31, 32, 33
<code>\bibcop@script</code> ....		
. 12, 16, 18, 19, 21, 26		