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

Yegor Bugayenko
yegor256@gmail.com

2025/03/10, 0.0.30

NB! You must run TeX 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 TeX 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 BibTeX 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 TeX 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 BibLaTeX is used), otherwise there won't be any warnings from `bibcop`.

*The sources are in GitHub at 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`, `@phdthesis`, `@incollection`, and `@misc` types of BIBTEX entries are allowed. Everything else, like `@manual`, `@inbook`, 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 BIBTEX entry. The tags that are not in the list are prohibited to use.

`doi` Every BIBTEX 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 tailing 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:

```
\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@noinproc--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

% \begin{macro}{\bibliography}
% Then, we re-define the |\bibliography| command:
%    \begin{macrocode}
\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

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.