



Tool for Customizing BibT_EX Style Files

Satoshi Hagihira

Department of Anesthesiology, Osaka University Graduate School of Medicine
2-2 Yamadaoka, Suita City, Osaka, Japan 565-0871
hagihira@anes.med.osaka-u.ac.jp

KEYWORDS BibT_EX, style file, bst, customization

ABSTRACT BibT_EX is a powerful tool to building reference lists from a bibliography database. Because the bibliography styling varies so widely among journals, a bibliography style file, capable of creating a list that exactly meets the requirements of a target journal, may not always be available. Since manually editing a BibT_EX style file to ensure compatibility is troublesome and prone to error, I developed cbst, a tool that employs shell scripting and gawk scripts to customize BibT_EX style files. Using cbst, it is possible to easily generate bibliographies that conform with the style of most target journal.

1 Introduction

There is wide and often subtle variation in reference-list styles indicated in guides for authors or in evidence from sample bibliographies. When a L^AT_EX class file and an appropriate BibT_EX style file are provided, authors can easily generate a correctly formatted reference list using BibT_EX and a bibliography database. However, when these resources are not available, because the structure of bibliography-style file is rather complicated and is written in a postfix stack language [3, 2], the task of manually editing a BibT_EX style file to generate a bibliography which is correctly styled for the target journal is usually time consuming and prone to error. For example, great care is needed to maintain the proper hierarchy of items that are pushed on to the stack. To simplify the process of custom bibliography generation, I developed a tool for editing BibT_EX style files.

2 Structure of a bibliography style file

First of all, with the aid of Patashnik's style guide [2], I analyzed the structure of the standard bibliography-style file (`plain.bst`). BibT_EX style files employ Reversed Polish Notation (RPN) [2]. Consequently, when analyzing commands written in the style file, particular attention must be given to the order of the commands. That said, a bibliography style file can be separated into several parts: declaration of variables; definitions of functions used in sub-routines; definitions of functions for each category entry (e.g., article, book, or inbook); and commands and macros.

When customizing the style file, there are several other important points. String variables can be defined by command `STRINGS` and numeric variables can be defined by command `INTEGERS`. Styling of the author's name is prescribed in the `'FUNCTION {format.names}'`, which can be modified by altering the formatting string, for example, `'{11}{~f}{~v}{~j}'` renders the name "Donald Ervin Knuth" as "Knuth DE". The terms and syntax of this format string are explained by Patashnik [2]. Briefly, '1' means last name, 'f' means forename name, 'v' means von name and 'j' indicates 'junior'. Any elements that are absent in the database name, they are ignored. Single letters prescribe abbreviation to an initial character and double letters prescribe full spelling. The user can add punctuation and other characters to a name string. For example, `'{11}{~f.}{~v}{~j}'` renders "Donald Ervin Knuth" as "Knuth D. E.". Note that, by default, a space is inserted between the initials or name strings: to override the insertion of a space, the user must add `{}`.

The format of each category entry is defined in the corresponding `bst` category function. For example, Figure 1 shows the `'FUNCTION {article}'` defined in `plain.bst`. If it is necessary to invert the order of author and title in the reference list, all you have to do is exchange the line containing the `format.title` command with the line containing the `format.authors` command.

As may be required by different journals, the `BIBTEX SORT` command can be applied to provide the reference list in order of appearance or in alphabetical order.

```

FUNCTION {article}
{ output.bibitem
  format.authors "author" output.check
  new.block
  format.title "title" output.check
  new.block
  crossref missing$
    { journal emphasize "journal" output.check
      format.vol.num.pages output
      format.date "year" output.check
    }
    { format.article.crossref output.nonnull
      format.pages output
    }
  }
if$
new.block
note output
fin.entry
}

```

FIGURE 1. A typical function, here for `article`, in `plain.bst`.

3 Specification of `cbst`

The purpose of `cbst` is to produce a bibliography-style file which generates reference lists that are fully compatible with the requirements of a target journal. The following four points are particularly relevant.

1. Styling of the author names.
2. Truncating the number of coauthors when the journal specifies a limited number per reference.
3. Listing, in proper order, author, title, journal, and other items in single reference entry.
4. Insertion of proper field separators such as spaced and unspaced semicolons and colons.

As mentioned in the previous section, the bibliography-style file can be separated into several parts. Accordingly, I divided `plain.bst` and `alpha.bst` into style-file fragments and added or modified relevant functions and variables for easier customization: six string variables, `pre`, `pre2`, `post`, `post2`, `separator`, and `str` were added. The first four are used to insert strings immediately before or after each item. `separator` defines how to mark the separation of field items. `str` is a general purpose variable. Because the number of string variables seems to be limited to 10, at most, 6 variables can be added to a style file based on `plain.bst` or `alpha.bst`.

I wrote two scripts to produce bibliography-style files. One is a gawk script, `mk-type`, which generates the `FUNCTION` of each category entry from the definition file. Figure 2 shows an extract from a typical definition file.

```
@article{
  author%", "
  nb
  title
  journal%emphasize
  separator%" "
  year
  separator%;" ;"
  volume
  separator%": "
  pages
  note
}
```

FIGURE 2. Extract from a typical definition file used by `cbst`.

As can be seen, the definition file is quite simple. The definition of each category is marked with `@(name of category)` immediately followed by a left brace. After the one-per-line listing of reference entry items, the category is terminated with a right brace. In this style file, the available category definitions are: `article`, `book`, `booklet`, `inbook`, `inproceedings`, `incollection`, `masterthesis`, `manual`, `phdthesis`, `proceedings`, `techreport`, `unpublished`, and `misc`. The user may omit definitions for any unnecessary categories. Between the braces, each line comprises an item name or command and, as applicable, one or two options. The field separator in each line was set as `'%`, because this character is never used as a field separator in reference lists. The order of items prescribes the order of appearance in the reference list. Separator prescribes the marker to be used for separating field items. For example, in Figure 2, a semicolon is specified as the field separator between year and volume. In the default setting, “and” is placed between

the penultimate and final listed author. The user can alter this by written in the second field of the author line. In the script extract in Figure 2, all author names are separated by a comma and space. On the next line, `nb` indicates a `new.block` command: this puts a period after the preceding item and starts a new line.

The user can also enter native style-file code in the definition file. If the first parameter is empty, `mktype` handles the second strings as native code and writes out them into `FUNCTION` without any modification. I made two specific commands: `missing` and `empty`. The second parameter of these commands is either item name or string `crossref`. These commands are used to effect conditional statements. These options enable offer more flexible coding.

Owing to apparent B_IT_EX buffer size restrictions, large style files cannot be loaded into memory. To avoid buffering problems, the `mktype` script was written parsimoniously to minimize the size of code and to eliminate redundancy, some `FUNCTIONS` that were defined in the original style file were deleted.

```
FUNCTION {article}
{ output.bibitem
  ", " format.author2 "author" output.check
  new.block
  format.title "title" output.check
  new.block
  journal emphasize format.str2 "journal" output.check
  " " 'separator :=
  year format.str2 "year" output.check
  ";" 'separator :=
  volume format.str2 "volume" output.check
  ":" 'separator :=
  format.pages2 output
  note output
  fin.entry
}
```

FIGURE 3. A ‘`FUNCTION {article}`’ generated by `mktype` from the definition file extract in Figure 1.

The other script, `cbst`, is a shell script that, through five dialogs, enables setting some of the styling requirements, such as for names, and finally constructs the complete style file from the separate pieces, which comprises all the generated functions for each category.

In the first dialog, the user is asked to select plain or alpha for the base style file. Then the author format is elicited. Next, the user is asked to set the maximum number of authors. After that, the user is asked to set the number of authors when the coauthor list has to be truncated. Finally, the user is asked to whether the reference list should be sorted or not. Once the dialog is completed, `cbst` will generate a style file.

Some journals restrict the number of listed authors. For example, the British Journal of Anaesthesia ‘Instructions to Authors’ states, “names and initials of six authors (if more than six, list three followed by *et al.*)” Because the original `plain.bst` had no functions for coauthor list truncation, I modified the ‘`FUNCTION {format.names}`’ using the `$if` command as shown in Figure 4. As shown, when there are more than five

coauthors, this function truncates the list to three coauthors and adds “, et al.” after the third.

```

FUNCTION {format.names}
{ 's :=
  #1 'nameptr :=
  s num.names$ 'numnames :=
  numnames 'namesleft :=
  % set max author
  numnames #5 >
  {
  % set max author +1
  #4 'namesleft :=
  { namesleft #0 > }
  { s nameptr "{11}{~f{}}{v{}}{, jj}" format.name$ 't :=
    nameptr #1 >
    { namesleft #1 >
      { ", " * t * }
      { ", et~al." * }
      if$
    }
    't
    if$
    nameptr #1 + 'nameptr :=
    namesleft #1 - 'namesleft :=
  }
  while$
  }
  {
  ...

```

FIGURE 4. Modified ‘FUNCTION {format.names}’, which is used when coauthor list truncation is required.

4 Discussion

The tool that I described here is easy to use and is flexible enough to enable modification of the bibliography-style file to generate a reference list that is fully compatible with target journal requirements. The user only has to set up a definition file in accordance with the journal’s instructions or normal practice. Developer of BibTeX Oren Patashnik has commented that the most frequent request he received was for new bibliography style files [4]. I developed this tool in the hope that BibTeX users would find it useful. At time of writing, the tool has been downloaded more than 4,500 times. Some cbst users have reported bugs or sent requests for added features or other improvements. Patrick Daly [1] has also developed the custom-bib/makebst package, which also enables the production of new bibliography style files. Using his tool, however, it is necessary to complete a series of 88 option dialogs. Each question requires a response before moving to the next one. While this is simple for the use, the procedure can be somewhat tedious. The tool I developed seems to be more flexible,

and more active involvement makes it easier to prescribe modifications to the bibliography style file. As shown in Figure 2, the structure of the definition file required for `mktype` is very simple, and can be easily modified using a text editor.

Of course, this tool is not perfect. In some cases, the user might still have to manually modify the generated bibliography-style file, but the modifications will be minor. The procedure is much easier than manual customization from scratch.

```
\bibitem[\protect\citeauthoryear{Antognini and Schwartz}\
{Antognini and Schwartz}{1993}]{MAC_brain_spinal}
Antognini, J.-F. and K.-Schwartz (1993).
\newblock Exaggerated anesthetic requirements in the preferentially
anesthetized brain.
\newblock {\em Anesthesiology}\vol{6}, 1244--9.
```

FIGURE 5. Extract of typical `bbl`-file output based on `chicago.bst`.

```
FUNCTION {label.out}
{
  author empty$
  { editor empty$
    { skip$ }
    { editor #1 "{ll}" format.name$ duplicate$ "}" * write$ "}" \
      * year * "}" * write$ }
    if$
  }
  { author #1 "{ll}" format.name$ duplicate$ "}" * write$ "}" \
    * year * "}" * write$ }
  if$
}

FUNCTION {output.bibitem}
{ newline$
  "\bibitem[\protect\citeauthoryear{" write$
  label.out
  cite$ write$
  "}" write$
  newline$
  ""
  before.all 'output.state :=
}
```

FIGURE 6. Modified ‘`FUNCTION {output.bibitem}`’ for `chicago.sty`.

Some journals require “Chicago-style” reference lists [5]. After inspection the contents of `chicago.bst`, I found that with only a minor modification, the generated style file can be adapted to Chicago style. Figure 5 shows a part of a `bbl` file made using the original `chicago.bst`. It differs from output processed by `plain.bst` in the header line. To generate this, as shown in Figure 6, I added the ‘`FUNCTION {label.out}`’ and modified the ‘`FUNCTION {output.bibitem}`’. Figure 7 shows a part of a `bbl` file made using the style-file manually modified for `chicago.sty`. The header part becomes

compatible with that shown in Figure 5. In ‘FUNCTION {label.out}’, I coded instructions to list only the first author in the header. If the user wants to obtain output that is fully compatible with that generated by `chicago.bst`, the user should replace these two functions written in the `chicago.bst`. To use this modified bibliography-style file, the user must include `chicago.sty` in the L^AT_EX source.

```
\bibitem[\protect\citeauthoryear{Antognini}{Antognini}{1993}]{\
  {MAC_brain_spinal}
  Antognini~JF, Schwartz~K.
  \newblock Exaggerated anesthetic requirements in the preferentially
  anesthetized brain.
  \newblock {\em Anesthesiology} 1993;79:1244--9
```

FIGURE 7. An example of output bbl-file by the style file modified for `chicago.sty`.

I also developed `jcbst`, a Japanese version of `cbst`. While it works well with the Japanese version of B_IB_TE_X (jB_IB_TE_X), the Japanese extension of jB_IB_TE_X has to handle both Japanese references and English references. This inflates the size of the bibliography-style file which might then exceed the buffer limit. When using `jcbst`, it is better to avoid complicated definitions.

The bibliography-style file is used only for producing the reference list. When preparing a manuscript, the way that citations are styled in the text also causes difficulties. Specific citation patterns are usually defined in the L^AT_EX class file. To resolve the issues, I wrote a shell script to customize `article.cls` and make it suitable for manuscript preparation. This script enables customization of the following: form of title page; location of page number; in-text citation style; and bibliography-header form. The user would be better to consider to modify the class file of L^AT_EX for manuscript preparation.

In conclusion, I developed a tool for customizing the B_IB_TE_X style file.¹ This tool makes it easy for users to produce bibliography-style files that generate reference lists that are fully compatible with the styling requirements of target journals.

References

1. Patrick W. Daly, *Customizing bibliographic style files*, 2003. CTAN:macros/latex/contrib/custom-bib
2. Oren Patashnik, *Designing B_IB_TE_X styles*, 1988. (The part of B_IB_TE_X's documentation that's not meant for general users.) CTAN:biblio/bibtex/contrib/doc/btxhak.pdf
3. ———, *B_IB_TE_Xing*, 1988. (Documentation for general B_IB_TE_X users.) CTAN:biblio/bibtex/contrib/doc/btxdoc.pdf
4. ———, *B_IB_TE_X yesterday, today and tomorrow*, TUGboat **24** (2004), no. 1, 25–30, <http://www.tug.org/TUGboat/Articles/tb24-1/patashnik.pdf>.
5. University of Chicago Press Staff, *The Chicago Manual of Style*, 15th ed., University of Chicago Press, 2003.

1. The tool can be downloaded from <ftp://masui.med.osaka-u.ac.jp/pub/public/latex/utis/cbst-1.01.tar.gz>. The full package, `manuscript-1.93.tar.gz`, which contains both `makecls` and `cbst` also can be downloaded from the same site.