bibtopic.sty

Pierre Basso
`Pierre.Basso@lim.univ-mrs.fr`

Stefan Ulrich
`stefanulrich@users.sourceforge.net`

2006/09/08 v1.1a

CTAN: `macros/latex/contrib/supported/bibtopic/`

Abstract

bibtopic.sty is a revised version of P. Basso's bibtopics.sty[1]. It allows for bibliographic references which are taken from several `.bib` files and written to separate bibliographies by LaTeX. These bibliographies may be considered as covering different topics (hence the name) or different material (such as references from articles, references from books, ...). The package works exclusively on the LaTeX side of things; it won't help your with constructing the the different `.bib` files for the various topics.[2]

Besides from offering features the former version didn't have, this version is compatible with various document classes and bibliographic styles (such as the KOMA classes and the natbib package). The package works with LaTeX2e only. It is currently maintained by S. Ulrich; please send bug reports or feature requests to `stefanulrich@users.sourceforge.net`.

Contents

---

[1] The name has been truncated to fit the 8+3 scheme of certain OSes.

[2] There are several tools that may help you with doing this; see section 7 for details about this.

Figures

1 Example of use

Here's a short example showing the main features of bibtopic.sty. Let's assume that you want to have two separate bibliographies in your document: References to *books* (with `books.bib` as bibliography file) and references to *articles* (with `articles.bib` as bibliography file). For brevity, `books.bib` shall contain entries for just two citation keys: `ColBenh:93` and `Munt:93`, and `Articles.bib` shall contain entries for the two keys `RouxSmart:95` and `Schwind:96`.[3] Then your input file might look as shown in figure 1 on page 3.

To produce a `.dvi` file, you'll have to run LaTeX and BibTeX as follows:

```
$ latex sample
... (message from bibtopic.sty to run BibTeX on 'sample1' and 'sample2') ...
$ bibtex sample1   (Note: Filename without extension!)
$ bibtex sample2
$ latex sample
... (message from LaTeX: 'Rerun to get references right') ...
$ latex sample
```

So there's one extra `.aux` file per `btSect`. These files have the same base name as your LaTeX source file, with an additional number that refers to the number of the `btSect` environment in the LaTeX file.[4] You need to run BibTeX on each of these files. The resulting `.dvi` file should look like figure 2 on page 4.

Some other things to note about this example:

- The second (the non-optional) argument of the `btSect` environment specifies the BibTeX source file(s) for the current 'topic'.
- Every `btSect` may use its own bibliography style (with some restrictions; see item 5 on page 12).
- The standard LaTeX command `\bibliograpy` doesn't work with bibtopic.sty, and is effectively disabled when using the package. Instead, the package has three separate commands `\btPrintCited`, `\btPrintNotCited` or `\btPrintAll` to print the citations:

---

[3] The `.bib` files are contained in the distribution, as well as all the other files mentioned in the example. So you may have a look at them while reading this, and run LaTeX/BibTeX on these files yourself.

[4] You can customize this naming scheme; see the description of the `\thebtauxfile` command, p. 5.

```
\documentclass[10pt]{article}
\usepackage{bibtopic}
\begin{document}
\bibliographystyle{alpha}

\section{Testing}
Let’s cite all the books: \cite{ColBenh:93} and
\cite{Munt:93}; and an article: \cite{RouxSmart:95}.

\begin{btSect}{books}
\section{References from books}
\btPrintCited
\end{btSect}

\begin{btSect}[plain]{articles}
\section{References from articles}
\btPrintCited

\section{Articles not cited}
\btPrintNotCited
\end{btSect}
\end{document}
```

Figure 1: Input file `sample.tex`

- `\btPrintCited` prints all references for the authors who have been cited in the document (for the references from `articles`, this is the article by *Roux and Smart*).
- `\btPrintNotCited` prints references for authors *not* cited in the document (*Schwind* in the example).
- `\btPrintAll` (not shown in the example) prints all references from the bibliography file, regardless of whether they have been cited or not.
- Unless the package option `printheadings` is specified, the default section titles and running headers of `thebibliography` are suppressed; the user may provide his own titles using LaTeX’s normal sectioning commands (`\section` in the example).

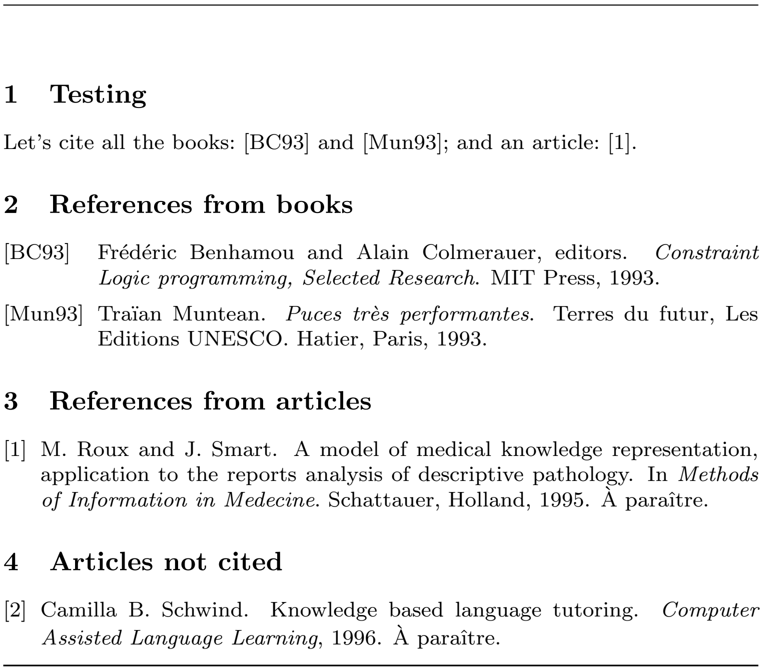
2 User interface

After this short example, let’s have a more detailed look on the syntax of the user commands.

2.1 Commands and environments

`\bibliographystyle{`⟨*style*⟩`}`

This command can be used anywhere (even several times) in the document to set the default bibliography style ⟨*style*⟩ for the subsequent `btSect` environments. Examples for ⟨*style*⟩ are `plain`, `named`, `agsm` etc. – anything for which



1 Testing

Let's cite all the books: [BC93] and [Mun93]; and an article: [1].

2 References from books

[BC93] Frédéric Benhamou and Alain Colmerauer, editors. *Constraint Logic programming, Selected Research.* MIT Press, 1993.

[Mun93] Traïan Muntean. *Puces très performantes.* Terres du futur, Les Editions UNESCO. Hatier, Paris, 1993.

3 References from articles

[1] M. Roux and J. Smart. A model of medical knowledge representation, application to the reports analysis of descriptive pathology. In *Methods of Information in Medecine.* Schattauer, Holland, 1995. À paraître.

4 Articles not cited

[2] Camilla B. Schwind. Knowledge based language tutoring. *Computer Assisted Language Learning*, 1996. À paraître.

Figure 2: Output from `sample.tex`

there exists a `.bst` file. (Note that some bibliography styles are incompatible with each other – see item 5 on page 12 for more about this.) So this command is quite similar to the standard LaTeX command with the same name, only that it can be used several times and that a default ⟨*style*⟩ (= `plain`) is used if it doesn't occur at all. The setting specified by this command can be overridden for specific `btSect`s by using `btSect`'s optional argument ⟨*style*⟩.

`\bibliography{`⟨*bib-files*⟩`}`

This command *cannot* be used together with bibtopic.sty; in fact, it is disabled and using it will result in a warning. Instead, use one of the commands `\btPrintCited`, `\btPrintNotCited` and `\btPrintAll` together with the `btSect` environment (described below).

`\begin{btSect}[`⟨*style*⟩`]{`⟨*bib-files*⟩`}`

This is the main environment of bibtopic.sty. It serves as a 'container' for the commands `\btPrintCited`, `\btPrintNotCited` and `\btPrintAll` that actually print the bibliographies. The environment determines the style and the BibTeX source file(s) for these references. Its arguments are:

⟨*style*⟩	optional argument to override the default `bibliographystyle` (see the `\bibliographystyle` command) for the scope of the current `btSect`.
⟨*bib-files*⟩	the filename(s) of the bibliography file(s) from which the references for the current `btSect` are taken. Just as in the standard `\bibliography` command of LaTeX, this argument consists of one filename or a comma-separated list of filenames without the `.bib` extension.

The following three commands can only be used inside a `btSect` environment. They read the `thebibliography` environment corresponding to the current ⟨*bib-files*⟩ and print some or all of the entries in it.

`\btPrintCited`
: Prints all references from ⟨*bib-files*⟩ that have been cited within the scope of the current `btUnit` environment (or in the entire document, if there is no `btUnit` environment – see the description of the `btUnit` environment below).

`\btPrintNotCited`
: Prints all references from ⟨*bib-files*⟩ for the citations occurring in ⟨*bib-files*⟩ that have *not* been cited in the current scope.

`\btPrintAll`
: The union of `\btPrintCited` and `\btPrintNotCited`: prints *all* references from ⟨*bib-files*⟩, no matter if they have been cited or not.

The following environment provides logically independent citation sections, i. e. it implements the behaviour you would get from using one of the chapterbib or bibunits packages (which are incompatible to bibtopic.sty).

`\begin{btUnit}`
: Normally, when you invoke one of the `\btPrint{Cited|NotCited}` commands, 'cited' means: A reference that has been cited somewhere in the entire document. However, if you enclose a part of your document into the `btUnit` environment, the scope of the citations is narrowed to this environment. This also means that you can have several `btUnits` containing the same bibliographic reference without getting a '`multiply-defined labels`' warning from LaTeX. An example input file using the `btUnit` environment is shown in figure 3, p. 6.

Note: When using this environment *all* of your citations have to occur inside some `btUnit`. Citations not in the scope of a `btUnit` won't work at all. Also note that this environment cannot be nested.

Finally, there's a command for customizing the names of the additional auxiliary files produced by this package – i. e. the `.aux` files written by `btSect`, and the `.bbl` files produced by running BibTeX on these `.aux` files:

`\thebtauxfile`
: The default definition of this command in bibtopic.sty looks like this:

```
\def\thebtauxfile{\jobname\arabic{btauxfile}}
```

So the basename of the additional auxiliary files is the basename of the LaTeX file (`\jobname`) with a counter `btauxfile` appended for uniqueness. This counter is incremented automatically by each `btSect` environment. This command may be redefined by the user at any place in the input file; such a redefinition will then hold for all subsequent `btSect` environments. For example, the `dot` package option redefines `\thebtauxfile` to include a dot to separate the filename and the number like this:

```
\renewcommand{\thebtauxfile}{\jobname.\arabic{btauxfile}}
```

```
\documentclass[10pt]{article}
\usepackage{bibtopic}
\begin{document}
\bibliographystyle{alpha}

\begin{btUnit}         %%% begin first btUnit
Let's cite all the books: \cite{ColBenh:93} and
\cite{Munt:93}; and an article: \cite{RouxSmart:95}.

\begin{btSect}{books}
\section{References from books}
\btPrintCited
\end{btSect}

\begin{btSect}[plain]{articles}
\section{References from articles}
\btPrintCited

\section{Articles not cited}
\btPrintNotCited
\end{btSect}
\end{btUnit}           %%% end first btUnit

\begin{btUnit}         %%% begin second btUnit
We may cite \cite{RouxSmart:95} another time without causing a
`multiple defined citations' warning from \LaTeX, since this
citation is located in another `btUnit'.

\begin{btSect}{articles}
\section{All articles}
\btPrintAll
\end{btSect}
\end{btUnit}           %%% end second btUnit

\end{document}
```

Figure 3: Example for using the `btUnit` environment (the output isn't shown).

Or if you want the package to use uppercase letters instead of digits, you may redefine `\thebtauxfile` in your document preamble like this:

```
\renewcommand{\thebtauxfile}{jobname.\Alph{btauxfile}}
```

You may also use any descriptive name for `\thebtauxfile`; but note that if you don't use the built-in counter `btauxfile`, you should double-check that each `\thebtauxfile` (for every `btSect`) does have a unique name; otherwise, a later auxiliary file with the same name as the current one would overwrite its contents, and bibtopic wouldn't work properly.

Also note that virtually any naming scheme may cause conflicts with existing `.aux` files – e. g., bibtopic will produce `foo1.aux` from `foo.tex`; if there also exists another file named `foo1.tex`, there is a name clash between the regular `.aux` file and the bibtopic file. However, this will probably only lead to serious problems when you're LaTeXing the files `foo.tex` and `foo1.tex` at the same time, since TeX doesn't lock its auxiliar files, and simultaneous read/write operations to the same file may cause a major mess.

2.2 Package Options

Here is an alphabetical list of all the options of bibtopic.sty:

`breakcites` Since the breakcites package is not compatible with the `btUnit` definition of bibtopic.sty (see section 4), this option may be used instead of the package to get the same behaviour.

`defaultbib` You should use this option as a workaround for the '`unknown bibliography`' error (see item 3 on page 9) in case you don't want to use some standard package with a 'known' definition such as the standard or KOMA classes or natbib.sty. This option will replace the (current) definition of `thebibliography` by some built-in default. In this case, a warning is given since the visual appearance of the bibliography might be different. This option does nothing if a known definition of `thebibliography` is used.

`dot` Changes `\thebtauxfile` to include an additional dot between the base file name and the number (`foo.1.`{`aux`|`bbl`} instead of `foo1.`{`aux`|`bbl`}; see the description of the `\thebtauxfile` command, p. 5 for details). This naming scheme was used in the old bibtopics.sty, so you may use the `dot` option for full compatibility with the old naming scheme.[5]

`normalwarnings` This option turns off the more eye-catching warning about outdated BibTeX files which is the default for bibtopic.sty. Normally, a command sequence triggering these warning messages is appended to the `.bbl` files included in the document (so that you can't forget or ignore the warning which will last until you relly rebuild the `.bbl` file); if you use this option, warning messages about outdated files ('`Package bibtopic Warning: xyz.bbl may be outdated`') will appear only for one LaTeX run, and warnings about nonexistant `.bbl` files will look more like the default warnigs ('`Package bibtopic Warning: No file sample1.bbl`'). The `.bbl` files are left alone. This option might be used to speed up LaTeXing and to save disk space with huge `.bbl` Files.

`overridenumbers/dontoverridenumbers` Some `.bst` styles store numbers for the bibliographic items in the optional argument of the `\bibitem` command, in conflict with continuous numbering of the `\bibitem`s for several `btSect` environments that is the default with bibtopic.sty. The package tries to catch this and re-activate its own counter, issuing a warning message. Using this option will disable the overriding behaviour (and the warning).

`printheadings` By default, the normal titles/headings for the bibliography section (`\bibname` or `\refname`) are suppressed when using bibtopic.sty. Use this option to print the headings as ususal.

`sectcntreset` Reset the numerical label for the bibliography items for each `btSect` environment to '1'. (The default is continuous numbering.) **Note:** This makes only sense when having at most one `\btPrintCited` command in all your `btSect` environments; otherwise, the citation labels in the document won't be unique any more! The bad thing is that you will get **no warning**

---

[5] The additional dot was omitted in the current version for compatibility reasons with certain OSes that don't support multiple dots in filenames.

from LaTeX about '`multiply-defined labels`' in this case – so please use this option with care.

`slow` By default each citation key is saved in the hash table of command sequences so that it can be checked efficiently by the `\btPrint...` commands. However, depending on the TeX implementation, the total number of command sequences may be limited; e. g. for teTeX 0.4, the limit is 9500, and in order to raise it, you'd need to recompile the binary. TeTeX versions > 0.9 allow for setting this limit dynamically, as do most other implementations (MikTeX, fpTeX).

When using this option, a linear list is used instead the internal hash table; this is slower, but will save hash space.

`unitcntnoreset` With this option, the bibitem counter will not be reset for for different `btUnits` (the default behaviour is to reset the counter at the beginning of a new `btUnit`).

`verbose` Give some messages about the `\bibliographystyle` used for the current `btSect`, and about `.bbl` files being marked as outdated. This may be useful if you get confused about which bibliography style is active at the moment.

2.3 Warnings and error messages

Most of the package-specific warnings should be pretty self-explanatory (such as the warning telling you to rerun BibTeX on certain files). As for error messages, we have tried to include explanatory messages (which can be invoked by typing '`h`' at the error prompt '`?`'); most of them will only occur under rare circumstances, so we won't explain them here. If you get some strange errors that you don't understand, please do also have a look at the section 'Bugs and Restrictions', p. 11.

In what follows, we'll explain some of the more non-obvious warning or error messages that you might encounter.

1. '`Rerun to get indentation of bibitems right.`'
   When using multiple bibliographies together with a numerical citation style, the item numbering will add up so that the space reserved for the items for each separate bibliography may be too narrow. This will break the left-alignment of the bibliography items. So we count the number of items printed in the bibliographies and use this number to set the items with correct spacing in the next run. As for all stuff done via the `.aux` file, an additional run may be necessary if the information written to the `.aux` file differs from previous runs; and this warning just tells you to do this additional run.

2. '`No appropriate bibitems found for command ... on line ...`'
   (probably together with an '`Empty 'thebibliography' environment`' warning). This will occur when a `\btPrint...` command tries to read entries that don't exist at all. This may be caused by

   - a previous error of some kind (outdated files, e. g.); in this case, just rerun BibTeX/LaTeX, and the warnings will disappear.
   - having included a `\btPrintCited` command for a bibliography file from which you didn't cite anything at all, or a `\btPrintNotCited` command

for a bibliography all references of which had been cited. The command name and the line number included in the warning message should help you with pinpointing the command that caused this warning. The `.dvi` file will just contain an empty space where you would expect the bibliography; so having separate headings for each of the bibliographies should help locating the erroneous command, too.

- a `thebibliography` enviroment using an unknown kind of `\bibitem`s (see also section 4 about compatibility with other packages). You should try to switch to another bibliography style in this case (or tell us of the incompatibility, if you really want to use the same style).

3. '`Found unknown thebibliography environment.`'
   This message means that the documentclass or some package used in your document provides a definition of the `thebibliography` environment that bibtopic.sty cannot handle (it has to change the headings, and, more important, parse the `\bibitem`s). The definition is evaluated as late as possible, at the beginning of the `btSect` environment; so any package in the preamble can influence the definition, regardless if it's loaded before or after bibtopic.sty. Probably the best solution for this problem is to use a 'knonw' bibliography style (the default or the KOMA classes, or natbib.sty should all work). If you can't do this, you can try to use the `defaultbib` option that will replace the offending definition with a built-in default (but it will warn you that such a replacement has happened, since the formatting could be different from what you would get without that option; see the description of the option on page 7 for more on this).

4. '`LaTeX Warning: There were undefined references.`'
   With the default referencing mechanism, you may get this warning when there have been changes in the bibliography; but in this case, the warning is always followed by another LaTeX warning saying '`Label(s) may have changed. Rerun to get cross-references right`'.[6] However, when you use the `btUnit` environment *without* using natbib at the same time, this second warning message won't show up (it will be suppressed by the definitions that prevent 'multiple label' warnings). This means that it's best to rerun LaTeX anyway to see if this warning message disappears. If it doesn't, then you'll have indeed undefined references in your document.

2.4 Obsolete Commands

The commands described in this section were part of the user interface of former versions of bibunits.sty. Since these versions are no longer officially supported, we strongly suggest that you don't use these commands for new documents: They don't share some of the features of the new commands and are provided for backward compatibility only. The descriptions here are intended mainly to help you with replacing them with the new commands.

`\begin{bibunit}[⟨style⟩]` Somewhat similar to `\btSect`, but the bibliography had to be specified by the `\putbib` command (which in itself is no longer

---

[6] So whenever this second warning does not show up, you know that you have indeed undefined references in your document.

supported, since it had been used exclusively in this environment). The actual bibliography was produced with a `\nocite{*}` command inside the environment, so the full syntax was:

```
\begin{bibunit}[⟨style⟩]
\nocite{*}
\putbib[⟨bib-files⟩]
\end{bibunit}
```

Instead of using these commands, you may now use the `\btSect` environment like this:

```
\begin{btSect}[⟨style⟩]{⟨bib-files⟩}
\btPrintAll
\end{btSect}
```

`\putbib[⟨style⟩]` See the `bibunit` environment.

`\begin{bibtopics}{⟨style⟩}{⟨bib-files⟩}` This command was an alias for the `bibunit` environment above:

```
\begin{bibunit}[⟨style⟩]
\nocite{*}
\putbib[⟨bib-files⟩]
\end{bibunit}
```

It can be replaced with a `btSect` as shown above.

3 Additional .aux files

For bookkeeping purposes, bibtopic.sty creates up to two extra files (apart from the files for each `btSect`): `btaux.aux` and `btbbl.aux`. The names have been chosen to minimize conflicts with existing (possibly important) user files; to further reduce this danger, bibtopic.sty adds a short ‘signature line’ (a comment) at the beginning of these files when writing them, and checks for the presence of this line before overwriting a file. If the file is non-empty, but the signature isn’t found, an error is raised: ‘`file ... not written by bibtopic`’. In this case, please check whether the file contains anything important, and if so, copy it to a safe place before continuing. (Alternatively, you could also rename the file, or move it to a place where TEX can’t find it.)

4 Compatibility with other packages

The package has been tested with current versions of the KOMA classes and current versions of natbib.sty.[7] Note that just like bibtopic.sty disables the usual bibliography section headings, it will also disable the ‘hooks’ provided by some packages/classes, e.g. for adding introductory or ‘preamble’ text to the bibliography (e.g. `\bibpreamble` in natbib.sty and current versions of KOMA). Instead of using these hooks, such text should be written directly inside the `btSect` environment.

In principle, bibtopic.sty should work with any document class or bibliography style, but there are some restrictions/exceptions:

---

[7] For KOMA and natbib.sty, the versions tested were 2.5c to 2.9 and 6.8c to 7.0, respectively.

- chapterbib.sty, bibunits.sty: These packages implement multiple, logically independent bibliographies. Since they do somewhat similar things (and work on the same commands) as bibtopic.sty, they are both incompatible with it. The `btUnit` environment should provide a similar functionality, so use this environment instead.

- hyperref.sty: When hyperref's `backref` option is used, bibtopic.sty has to be loaded *after* hyperref. (The user is warned if this is not the case.) The `backref` option loads the package backref.sty, and bibtopic.sty has been tested with backref.sty versions 1.16 to 1.19 (1999/04/12–2000/01/19).

- Packages redefining the `thebibliography` environment in a non-standard way: These definitions may cause the parsing mechanism of the `\bibitems` to fail. See the discussion of the '`unknown bibliography`' error message (item 3 on page 9) and the `defaultbib` option (page 7) for more about this.

- Packages redefining the `\@citex` command (like, e. g. breakcites) won't work together with bibtopic.sty when the `btUnit` environment is used. Since bibtopic.sty redefines `\@citex` as late as possible (inside `btUnit`), it's most likely that the definitions of the other packages will be overwritten. The case with breakcites.sty is dealt with by the `breakcites` option; if you observe this behaviour with other important packages, please email us about them.

- Packages or `.bst` styles using non-standard names for bibliography items (other than `\bibitem` or `\harvarditem`). In this case, parsing the `\bibitems` will also fail, and will result in an empty bibliography. These cases would have to be treated explicitly inside bibtopic.sty, so if you know of important packages that have these non-standard item commands, please email us about them.

- harvard.sty: We withdrew compatibility with this package in favour of the current versions of natbib.sty which has just the same features; so use natbib instead.

- Some `.bst`-Styles for numerical styles (e.g. `plaindin.bst` by K. F. Lorenzen) incorporate the item number into the optional argument of `\bibitem`; hence, the internal counter won't be used and bibtopic.sty would not be able to manipulate the counter for continuous numbering. bibtopic.sty tries to recover in this case by ignoring the optional numerical argument and giving a warning. This warning can be turned off by using one of the package options `overridenumbers` or `dontoverridenumbers` (see p. 7 above).[8]

5 Bugs and Restrictions

1. BibTeX's cross-referencing doesn't work between items in different `btSects`. Since BibTeX is run separately on the files corresponding to different `btSects`, it won't be able to resolve the cross-reference.

---

[8] Another possibility is to change the output function of the `.bst` file; see the source file `bibtopic.dtx`, lines 1387ff. for an example.

2. When the bibliography files have several items with the same author and the same year, they are tagged with ‘a’, ‘b’, ‘c’ etc. extensions even if not all of them are cited. When the bibliography is printed with `\btPrintCited`, funny effects might occur, e.g. a bibliography where only a ‘b’ item is shown.

3. bibtopic.sty doesn’t work with the ‘unsorted’ citation styles such as `unsrt` or `unsrtdin`.[9] Use the package multibib.sty instead (see also section 8).

4. bibtopic.sty is case sensitive, while BibTeX isn’t. That means that while BibTeX treats a `.bib` entry like:

   `@article{Gnus:98, ...`

   and a citation `\cite{gnus:98}` as the same key, bibtopic.sty treats them as different keys, and you will get an ‘`undefined reference`’ error for the citation command.[10]

5. When mixing several citation styles, it’s important to know that numerical citation styles and author-year styles are generally *incompatible* with each other. This means that the style declared first will always override later redeclarations, since it is globally set at the begin of the document and can’t be changed afterwards. Examples for numerical styles in that sense are `plain` and `alpha` (even if the name doesn’t sound like it), examples for author-year styles are the ‘harvard’ styles `agsm` and `dcu` or the `authordate1-4` style. As for these examples, you may mix `alpha` with `plain`, but not with `agsm` or vice versa.

6. Having the same reference in several bibliographies in the same `btUnit` (e. g., when using both the `\btPrintCited` and the `\btPrintAll` command to print parts of the same `thebibliography` twice) may lead to problems with labelling:

   With the numerical or ‘plain’ labelling styles, such multiple references are treated as *different keys*, i.e. they receive different numbers.

   With natbib.sty, things are a bit different: Multiple references will get the same number; but the numbering will have *gaps* since the repeated entries will be assigned the number of the *last* `\bibcite` enty of the `.aux` file of that repeated entry. You will also get warnings about ‘`multiply defined citations`’, and the warning ‘`Rerun to get citations correct`’ will appear after each LaTeX run.

   The latter problem can be avoided by switching to alphabetical or author-year styles, but the warnings about ‘`multiply defined citations`’ will remain. To get rid of these, you would need to use `btUnits` and put the `\btPrintAll` and the `\btPrintCited` command into different `btUnits` (see figure 3 on page 6 for an example).

---

[9] bibtopic.sty writes global `\citation{*}` commands to its additional `.aux` files: When constructing the `.aux` file we don’t know in which of the possibly several `.bib` files some citation will occur. So since there are no specific `\citation` commands in the (additional) `.aux` files, BibTeX can only apply author-year sorting or print the items in the order they appear in the `.bib` file.

[10] The reason for this restriction is that BibTeX reads the `\citation` labels when reading the `.aux` file and writes out the keys for the `\bibitem` commands it produces accordingly. However, bibtopic only writes a `\citation{*}` to the `.aux` file, so BibTeX can’t access these labels and therefore has to use the spelling from the `.bib` file.

7. Switching from one bibliographic style file to another may cause compiling errors (due to the outdated `.aux` and `.bbl` files). Try to make LaTeX ignore these errors by responding '`r`' after the help prompt '`?`' (don't quit with '`x`', because you want the `.aux` file to be regenerated); if all fails, delete the `.aux` and `.bbl` files before the next LaTeX run.

8. Brackets in the `.bib` file: The brackets '`[]`' are special to LaTeX in that they denote the bounds of an optional argument, and `\bibitem` uses such an optional argument for the item labels. Therefore using brackets '`[...]`' in a `.bib` file entry which is used for sorting may mangle the `\bibitem` command. In this case, hide the brackets from LaTeX by enclosing them in another pair of braces, like this: '`{[...]}`'. (bibtopic.sty will try to detect such misleading brackets an warn you about them.)

6 If you're not a user of BibTeX . . .

. . . (which would be a pity, though), the feature of selecting only those items actually cited in the document might come in handy: All you'll have to do is provide a sorted `thebibliography`, and bibtopic.sty will be able to select the needed citations. The easiest way to use this is to put your `thebibliography` environment into a separate file (call it `mybib.bbl`) that can be included using the standard `btSect` environment like, e. g.:

```
\renewcommand{\thebtauxfile}{mybib}
\begin{btSect}{}
\btPrintCited
\end{btSect}
```

In this case, the second argument of `btSect` serves no purpose and may as well be left blank. It'll be also a good idea to turn off persistent warnings from bibtopic.sty by using the `normalwarnings` option – else, an additional 'warning line' might be appended to your `mybib.bbl` in the first LaTeX run: bibtopic.sty will realize that the `.aux` file has changed, and it will tell you to rerun BibTeX to update your `.bbl` File. **Don't do that:** Running BibTeX as advertised would overwrite your handwritten `.bbl` file with an *empty* bibliography! With the `normalwarnings` option, this message won't occur; otherwise, you'd have to delete the 'warning' line from `mybib.bbl` manually in order to get rid of these recurrent warnings.

7 Other tools that might be useful with bibtopic.sty

You'll probably need to split up your exitsting `.bib` file(s) when switching to bibtopic.sty. This section lists some utilities that might be useful for this task. Some of these tools can be used for other purposes, too – like sorting, merging, consistency checks, automatic key generation etc.; but here we'll concentrate on the extraction issue only.

Note that if you want to keep all your references together in one master `.bib` file, you can always use a dummy attribute that will be ignored by BibTeX but may be used as a key for selecting entries with one of the tools mentioned here.

- BibTool (available as C source code on CTAN: `biblio/bibtex/utils/bibtool/`). It allows you to specify regular expressions for extracting entries from a `.bib` file. For example, the following command extracts all 'online documents' (items containing the string `http:` or `ftp:` in any field) from `mybib.bib` and writes them to a newly created bibliography file `online.bib` (the single quotes are needed with some Unix shells to prevent unwanted expansion of braces):

  ```
  bibtool -- 'select{"[hf]t?tp:"}' mybib.bib -o online.bib
  ```

- btOOL (C/Perl library, available from: `biblio/bibtex/utils/btOOL/`; more documentation can be obtained from the author's home page: `http://starship.python.net/~gward/btOOL/`). A powerful library for writing Perl scripts that manipulate `.bib` files. Very flexible, but requires Perl programming skills.

- `bibextract` (shell and awk scripts, available from: `biblio/bibtex/utils/bibextract/`). Similar to BibTool, it lets you use regular expressions to extract entries from a `.bib` file. With `bibextract`, the example above would look like this:

  ```
  bibextract "" "[hf]t?tp:" mybib.bib > online.bib
  ```

8 Related Packages

There are several packages with a similar purpose as bibtopic.sty:

multibib.sty: This package provides similar functionality as bibtopic, but with a different approach for the user interface: Instead of splitting up the `.bib` files, you use different, bibliography-specific citation commands. In contrast to bibtopic.sty, multibib.sty can also deal with '`unsorted`' bibliography styles (see page 12 above).

chapterbib.sty, bibunits.sty: These provide multiple bibliographies as well; the main difference is that with these packages different parts of a document can have their own dedicated bibliography which is independent from the other bibliographies, whereas bibtopic/multibib provide a global bibliography divided into sections.

Camel: Along with many powerful additional features, this larger package also contains macros for sectioned bibliographies; the user interface is quite different from standard LaTeX (e. g., the `\cite` command is replaced by a `\source` command).