Other cross-referencing tools for LaTeX

Richard Kaye
School of Mathematics and Statistics
The University of Birmingham
Birmingham B15 2TT
U.K.

12th August 1998

Contents

1 Introduction

This gives examples for using supplementary TeX programs, BibTeX a bibliography database tool, and `makeindex`, an index generator.

2 BibTeX

2.1 Why use BibTeX?

There are several reasons.

- Publishers are very fussy about how references and bibliographies are formatted, usually with good reason. If you use BibTeX correctly, you can make any required stylistic changes automatically, and you just have to concentrate on the content of your piece.
- BibTeX automatically sorts the references according to whatever scheme you require.
- You may build a database of the papers you refer to often, for use in your next paper or book.

In the long run, BibTeX saves time, but there are disadvantages.

- It takes more effort and requires more care to write the input file for BibTeX.
- Cross-references are not collated by LaTeX until at least the third pass on your main file.

2.2 How to use BibTeX

The general scheme is as follows: you create a LaTeX file as before, with the `\cite` commands, but leaving off the bibliographic data. You create one or more BibTeX files (which have extension `.bib`) and instead of the

```
\begin{thebibliography}...\end{thebibliography}
```

you use a

```
\bibliography{file1,file2}
```

command, which will instruct BibTeX to read files `file1.bib` and `file2.bib` and put the bibliography at that point. You will also need a

```
\bibliographystyle{plain}
```

command just after the `\begin{document}` command.

The data is collected in four stages as follows.

- LaTeX reads your main file and makes a note of all citations that are required. (This data goes in the `.aux` file.)
- BibTeX reads the `.aux` file and then reads any required `.bib` files, and creates a `.bbl` file containing all the references required in a format that LaTeX can read.
- LaTeX reads the `.bbl` file on the second run, when it encounters the command '`\bibliography`'. Note that this is usually at the end of the document. The `.bbl` file also contains the printable keys, and these are written to the `.aux` file at this stage.
- LaTeX is run again, and on this time will it be able to supply the appropriate keys in the final printed text.

Thus for a LaTeX document `foo.tex` using BibTeX, the following is the minimum you will need to do to resolve all cross-references.

```
latex foo
bibtex foo
latex foo
latex foo
```

In particularly bad situations, a further run of LaTeX is required. I have even come across `.bib` files which cross-reference other citations, and here a further `bibtex foo`, `latex foo`, `latex foo` were required.

2.3 Examples of the use of BibTeX

The .bib file is set out in a different format to normal LaTeX files. An example is given in xampl.bib. This example is designed to show you almost all of BibTeX's features. In practice, you won't use them all, so here's a quick summary or two entries in xampl.bib

The first is the spurious paper by Aamport on a parallel document preparation system [1].

```
@ARTICLE{article-full,
   author = {L[eslie] A. Aamport},
   title = {The Gnats and Gnus Document Preparation System},
   journal = {\mbox{G-Animal's} Journal},
   year = 1986,
   volume = 41,
   number = 7,
   pages = "73+",
   month = jul,
   note = "This is a full ARTICLE entry",
}
```

Here are some points to note.

- Names are usually written in full, 'Leslie A. Aamport'. Writing them 'Aamport, Leslie A.' is an alternative. The square brackets are not really needed. They were used here to indicate that the author's name *really is* Leslie, but this doesn't appear in the article. I don't recommend square brackets. Many BibTeX style will abbreviate the first given name to an initial anyway, and this is what is usually required.

- The title of the article is written fully capitalized. Most BibTeX styles will remove the capitals for articles (except for the very first word), but not for books. This serves as a useful way of distinguishing the two. If a word in the article's title really should be a capital, you put it in braces, so an article title might appear in a .bib file as

  ```
  title={The Size of {E}instein's Brain}
  ```

  and the E will always remain a capital letter.

- Double quotes "..." and braces {...} are both possible ways to delimit fields. It doesn't matter what you use. The commas between field are important though.

- Some entries do not require braces or double quotes. These are entries which are numbers, or are a predefined string, such as 'jul' here.

- Page numbers are usually wriiten as pages={23--34} (but you can use - instead of --, BibTeX automatically changes - to -- here).

- Some of the information is not required, other parts are necessary. You will get an error message from BibTeX if you miss out something important.

- ‘`\mbox`’ is the correct LaTeX way of ensuring a word is not hyphenated and split across two lines.

Here’s a ‘book’ entry [2].

```
@BOOK{book-full,
   author = "Donald E. Knuth",
   title = "Seminumerical Algorithms",
   volume = 2,
   series = "The Art of Computer Programming",
   publisher = "Addison-Wesley",
   address = "Reading, Massachusetts",
   edition = "Second",
   month = "10~" # jan,
   year = "{\noopsort{1973c}}1981",
   note = "This is a full BOOK entry",
}
```

Note `#`, used to concatenate two strings together, and `\noopsort` to fool BibTeX into thinking that the book was actually published in 1973, so that it is ordered as such when the refenences are put into order. (But both these features are fairly ‘advanced’.)

The other type of entry I use a lot is ‘incollection’. Here is an example [3].

```
@INCOLLECTION{incollection-full,
   author = "Daniel D. Lincoll",
   title = "Semigroups of Recurrences",
   editor = "David J. Lipcoll and D. H. Lawrie and A. H. Sameh",
   booktitle = "High Speed Computer and Algorithm Organization",
   number = 23,
   series = "Fast Computers",
   chapter = 3,
   type = "Part",
   pages = "179--183",
   publisher = "Academic Press",
   address = "New York",
   edition = "Third",
   month = sep,
   year = 1977,
   note = "This is a full INCOLLECTION entry",
}
```

Note the way you specify multiple authors, with an ‘and’ between each. BibTeX will make this into something more suitable. (What it chooses depends on the style you are using.)

3 *Makeindex*, an index processor for LaTeX

3.1 When and why to use *Makeindex*

Making good indices is a real pain in the neck. There is a lot of work collecting data, sorting it, and formatting it. Each of these tasks involves a lot of repetious work, but each also requires human judgement too. Use of a computer can reduce some of the labour, but unfortunately not a lot.

Do not start to think about making an index until your document is complete or incredibly nearly so.

3.2 How to use *Makeindex*

1. Put the command '`\usepackage{makeidx}`' in the preamble of your document, after the `\documentclass` command, before `\begin{document}`.
2. Put '`\makeindex`' between the `\usepackage{makeidx}` command and the command `\begin{document}`.
3. Put '`\printindex`' where you want the index to go—usually just before the `\end{document}`.

A `.idx` file will be created, with all the index entries. This file will contain the index item and the page number for every `\index` command in the document. For example, if your document says `\index{number theory}` and this occurs in page 27, then the `.idx` file will contain this information and an entry will be made in the index.

The `.idx` file is not read directly, but sorted and collated to produce a `.ind` file which actually produces the index. To get this file the process is

```
latex foo
makeind[e]x foo
latex foo
```

Note, on some computers (such as MS-DOS machines) *Makeindex* is actually called `makeindx` because these machines limit commands to eight letters. On unix you type the full word.

This document has various `\index` commands scattered around it. There are some variant of the commands, for example

```
\index{makeindex!useof}
```

creates a sub-entry 'use of' for the entry 'makeindex' (you can have subsubentires too);

```
\index{makeindx|see{makeindex}}
```

creates a cross-reference; and in

```
\index{alpha@$\alpha$}
```

The index entry is given as α but placed in the appropriate order as if it had been spelt 'alpha'.

See the documentation for *Makeindex* for more details.

3.3 How to use *Makeindex* to create a glossary

The commands `\makeglossary` and `\glossary` behave rather like the commands `\makeindex` and `\index`, but work with a `.glo` file instead. You can use the program *Makeindex* to sort these entries if you like, but formatting is usually more difficult and dependent on the type of application you have. I'm not going to provide clues here, but if you are interested, please ask. I did once use the LaTeX 'glossary' commands to produce an index of notation for a book once, and this was pretty successful. (In this particular case I chose not to use *Makeindex* to sort the entries though.)

References

[1] L[eslie] A. Aamport. The gnats and gnus document preparation system. *G-Animal's Journal*, 41(7):73+, July 1986. This is a full ARTICLE entry.

[2] Donald E. Knuth. *Seminumerical Algorithms*, volume 2 of *The Art of Computer Programming*. Addison-Wesley, Reading, Massachusetts, second edition, 10 January 1981. This is a full BOOK entry.

[3] Daniel D. Lincoll. Semigroups of recurrences. In David J. Lipcoll, D. H. Lawrie, and A. H. Sameh, editors, *High Speed Computer and Algorithm Organization*, number 23 in Fast Computers, part 3, pages 179–183. Academic Press, New York, third edition, September 1977. This is a full INCOLLECTION entry.

Index