chapterbib
multiple bibliographies in LaTeX

Donald Arseneau
asnd@triumf.ca

2010/09/05

Introduction

The chapterbib package facilitates multiple bibliographies in a LaTeX document, including items `\cite`d (cited) in more than one bibliography. Despite the name 'chapterbib', the *bibliographies are for each included file*, not necessarily for each chapter, although a bibliography per chapter is the usual application. The main point is to allow you to use BibTeX: Each included file should have its own `\bibliographystyle` and `\bibliography` commands, and you should run bibtex on each included file separately rather than on the main or root file.

Chapterbib also provides the environment `cbunit`, and the command `\cbinput` to allow multiple bibliographies without using `\include` (see item 3). There are two added hooks, `\citeform` and `\citepunct`, which you can redefine to customize the formatting of each entry in a citation list, and the declaration `\CitationPrefix` to use in preference to `\citeform` for numbering-by-chapter.

Alternative packages: bibunits, biblatex.

Usage, Restrictions, and Options

1 Normal use: Put `\bibliographystyle` and `\bibliography` commands in each `\included` file. Run LaTeX; run BibTeX on each included file; run LaTeX; run LaTeX.

2 Whole bibliography: With chapterbib, the `\bibliography` and `\bibliographystyle` commands are not normally used in the root file, only in files that have been `\included`. To have a whole-document bibliography, see items 6–9, depending on which style of whole-document bib.

3 Without \include: If you can't use `\include` because a new section must start below the preceding bibliography on the same page (odd format!), then you can use `\begin{cbunit}`...`\end{cbunit}` (for everything in one file) or `\cbinput`, with a `thebibliography` environment in each unit or input file.

To use BibTeX, input separate files using `\cbinput`; at first use the package or global option `[draft]`, run LaTeX on the document, then BibTeX on each file that was `\cbinput`; finally, remove the `[draft]` option and run LaTeX again (maybe twice to get page references right). The `[draft]` option only affects the treatment of `\cbinput`, not `\include` or `\begin{cbunit}`.

4 Package compatability: Your preferred citation style (call it xxx.sty) may not work with chapterbib at first, but it is easy to make it compatible: In 'xxx.sty' change every '`@\@citeb`' to '`@\@citeb\@extra@b@citeb`', and insert the line

```
\@ifundefined{@extra@b@citeb}{\def\@extra@b@citeb{}}{}
```

somewhere (but not as a comment or as part of another definition!).

If the package also redefines `\bibcite` then you should change that definition, replacing '`@#1`' with '`@#1\@extra@binfo`', and insert

```
\gdef\@extra@binfo{}
```

somewhere in the file. If the package defines a command that acts similarly to `\bibcite` (being written to the aux file, and then executed as the aux file is processed), then it should have '`\@extra@binfo`' inserted in the same way.

Some citation packages deviate quite far from LaTeX's own method of organizing cite tags using '`b@\@citeb`'. The instructions above catch such extensions as '`Y@\@citeb`', but not more radical differences.

In such cases, try contacting the author of the citation package.

If a citation style does not (re)define `\nocite`, then that command would not be converted when you make the patches at '`@\@citeb`'. Chapterbib will try to detect the presence of `\@extra@b@citeb` in `\nocite` and insert it, but if that fails you may need to redefine `\nocite` changing any '`@\@citeb`' to '`@\@citeb\@extra@b@citeb`' in that sty file.

5 Sectionbib: The report and book document classes usually treat the bibliography as an unnumbered chapter (`\chapter*`), which is not so good for bibliographies *in* a chapter. You can specify

```
\usepackage[sectionbib]{chapterbib}
```

to convert your bibliographies from `\chapter*` to `\section*`, with an entry in the table of contents and the page-header. A bibliography in the root file remains as a `\chapter*`. The `[sectionbib]` option modifies the existing `thebibliography` environment (or the `\bibsection` command, if present already), so the other formatting in the bibliography should remain unchanged. On the other hand, if you already have a non-standard bibliography defined, or if you want them numbered, it may be easier to redefine `\thebibliography` directly, without any tricky modification of existing commands.

Alternatively, you can use the `\sectionbib` command directly in the document preamble. It takes two parameters: the sectioning command, and the name of the sectioning level. For instance, the `[sectionbib]` option executes `\sectionbib{\section*}{section}`. Again, for the most control, it is better to redefine `\thebibliography` entirely.

6 Overall separate bibliography: If you want a completely unrelated bibliography in the root file, perhaps for a general reading list, you can provide your own bibliography there using the `thebibliography` environment. I don't suppose this will appeal to BibTeX users!

7 Overall bibliography: To have a cohesive bibliography for the whole document, plus individual bibs in the chapters, put `\bibliography` commands in the included chapters plus in the root file. Make sure the `\bibliographystyle` for the overall bibliography appears *first*, before any chapters are included. Run LaTeX; run BibTeX on the root file; run BibTeX on each included file; run LaTeX; run LaTeX. This produces an independent 'overall' bibliography which

only makes sense for various 'named' bibliography styles; a numbered style, or one with any type of automatic enumeration (like Me2007a, Me2007b) will give unrelated numbers in each bibliography and lead to confusion.

BibTeX will complain about multiple `\bibdata` commands when it makes the whole bibliography, but it should obey the first. If you don't want to see any error messages from bibtex, or if you don't want to put the main `\bibliographystyle` command first in the document, then use `\usepackage[rootbib]{chapterbib}` when you run LaTeX first; run BibTeX on the root file; change to `\usepackage{chapterbib}`; run LaTeX; run BibTeX on each included file; run LaTeX; run LaTeX.

8 Chapter bibs gathered to end: To have a bibliography-by-chapter at the end instead of separate bibs in the chapters, use `\usepackage[gather]{chapterbib}`, put `\bibliography` commands in each file, and at the end of the main file. Run LaTeX as in item 1. You can control the titling of the final bibliographies by defining `\FinalBibTitles`, such as

```
\newcommand\FinalBibTitles
   {References for Chapter \thechapter}
```

A similar effect may be achieved by re-defining `\FinalBibPrefix` as

```
\renewcommand\FinalBibPrefix{References for }
```

Even more control is achieved by redefining `\StartFinalBibs`. The default definition is (like)

```
\newcommand{\StartFinalBibs}{%
  \renewcommand{\bibname}{Bibliography for chapter n}}
```

normally, but when using the [sectionbib] option it becomes

```
\newcommand{\StartFinalBibs}{\chapter*{\bibname}%
  \addcontentsline{toc}{chapter}{\bibname}%
  \@mkboth{\bibname}{\bibname}%
  \renewcommand{\bibname}{Chapter n}}
```

where the `\bibname` text is now provided by `\@auto@bibname`, which relies on bookkeeping and `\FinalBibPrefix`.

If your document class has neither section nor chapter, then you must define `\StartFinalBibs` and also indicate the sectioning: for example, if the main sectioning command in your document class is `\motif`:

```
\newcommand\CBMainSectioning{motif}
```

9 Duplicate bibliographies at end: To have bibliographies in each chapter *plus* a bibliography-by-chapter at the end, follow item 8, but declare

```
\usepackage[duplicate]{chapterbib}
```

(or `\usepackage[duplicate,sectionbib]{chapterbib}`).

10 Babel: If you use Babel, load chapterbib before babel.

Formatting extensions

\citeform Normally, the citations are formatted as given, but you can define `\citeform` (with one parameter) to reformat every citation. Some possibilities:

```
\renewcommand\citeform[1]{\romannumeral 0#1}          [iv,x]
\renewcommand\citeform[1]{(#1)}                       [(3),(4),(7)]
```

If you change `\citeform`, you should really define `\@biblabel` to match.

A not-so-good way to provide a chapter-number prefix is

```
\renewcommand\citeform[1]{\thechapter.#1}
```

This partially works, but has only limited applicability: it does not work with cites in the front-matter (TOC, LOF) or with hyperref.

\CitationPrefix Instead, there is a `\CitationPrefix` command to apply a prefix to the citation numbers (or names) in the bibliographies and `\cite` commands for the included files. Use it by declaring something like

```
\CitationPrefix{\thechapter.}
```

in the preamble. The prefix will be applied to all the chapter-bibs but will not be used in an overall (root) bibliography, if you have one. This not only affects the formatting but the actual citation, therefore it immediately applies to `\bibitem` (no need to change `\@biblabel`) and works with hyperref. (The `\CitationPrefix` mechanism may be prone to conflicts with other packages.)

\citepunct The `\citepunct` command gives the punctuation (comma-penalty-space) between items in the `\cite` list, and you can redefine it.