The **chapterbib** package*

Donald Arseneau

31–Mar–2009

Contents

---

*This manual corresponds to chapterbib v1.14, dated 31–Mar–2009.

1 List of Changes

	Version	
1.14	(31–MAR–2009)	DA (\nocite*, \CitationPrefix, etc)
1.13	(11–DEC–2007)	DA (new final-bib titling)
1.12	(21–MAR–2004)	DA
1.11	(29–FEB–2004)	DA (allow \nocite check)
1.10	(23–JUN–2003)	DA (\bibsection \CBMainSectioning)
1.9	(19–SEP–2001)	DA (sectionbib change; babel)
1.8	(29–APR–1999)	DA (gather, duplicate, toc)
1.7	(21–JUL–1997)	DA (sectionbib, \nocite)
1.6	(08–FEB–1997)	Donald Arseneau (more sectionbib)
1.5	(09–OCT–1995)	Donald Arseneau (rootbib)
1.4	(11–MAR–1995)	Donald Arseneau (sectionbib)
1.3	(04–JUL–1994)	Donald Arseneau (2e, cbunit)
1.2	(21–MAY–1993)	Donald Arseneau (bug fix)
1.1	(24–MAR–1993)	Donald Arseneau
1.0	(23–NOV–1988)	Niel Kempson

2 User Guide

Allow multiple bibliographies in a LaTeX document, including items \cite'd in more than one bibliography. Despite the name "chapterbib", the BIBLIOGRAPHIES ARE FOR EACH INCLUDED FILE, not necessarily for each chapter. 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 root file.

There are also the commands \begin{cbunit}, \end{cbunit}, and \cbinput to allow multiple bibliographies without using \include (see item 4). There are two added hooks, \citeform and \citepunct, which you can redefine to customise the formatting of each entry in a citation list, and the declaration \CitationPrefix to use in preference to \citeform.

2.1 Usage, Restrictions, and Options

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

2. The \bibliography and \bibliographystyle commands are not normally used in the root file, only in files that have been \include'd. To have a

whole-document bibliography, see items 6–9, depending on which style of whole-document bib.

3. 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}` 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}`.

   With old LaTeX, do the preliminary run using `\include` commands, and change these to `\cbinput` for the final run(s).

4. Your preferred citation style (`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.

   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 define "`\nocite`", then that command would not be converted when you make the patches to "`@\@citeb`". Chapterbib will try to detect the hook in "`\nocite`", but if this fails you may need to redefine "`\nocite`" (with any "`@\@citeb`" changed to "`@\@citeb\@extra@b@citeb`") in that sty file.

5. 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 trickery.

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 does `\sectionbib{\section*}{section}`. Again, for the most control, it is better to redefine `\thebibliography` entirely.

6. 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. 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. 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 achived 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. To have bibliographies in each chapter PLUS a bibliography-by-chapter at the end, follow item 9, but declare `\usepackage[duplicate]{chapterbib}` (or `\usepackage[duplicate,sectionbib]{chapterbib}`).

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

2.2 `\citeform`, `\citepunct` and `\CitationPrefix`:

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}% roman numerals: [iv,x]
\renewcommand\citeform[1]{(#1)} % parentheses: [(3),(4),(7)]
```

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

Another not-so-good example to provide a chapter-number prefix is

```
\renewcommand\citeform[1]{\thechapter.#1} % number by chapter.
```

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

`\citepunct` gives the punctuation (comma-penalty-space) between items in the `\cite` list.

3 Implementation notes

LaTeX normally uses command names in the form `\b@⟨TAG⟩` to associate a mnemonic ⟨*TAG*⟩ with the citation name or number. Chapterbib changes this to a command like `\b@⟨TAG⟩@-⟨nn⟩` where ⟨*nn*⟩ is a number identifying which included file is in effect.

3.1 Tags indicating the citations and/or the input files:

`\c@inputfile`   = counter counting included files

`\the@ipfilectr` = (empty) when typesetting from the root file,

= `@-\the\c@inputfile` when typesetting from an included file

`\@extra@b@citeb` = `\the@ipfilectr` (just an alias)

`\@extra@binfo`   = the value of `@extra@b@citeb` as saved in `.aux` files

3.2 `\b@⟨FOO⟩`:

In the root file, the citation number (or name) is given by `\b@⟨FOO⟩` just like regular LaTeX, but in an `\include`'d file it is given by `\b@⟨FOO⟩@-⟨number⟩`. Any definition of `\cite` (`\@citex`) should refer to this using
`\csname b@\@citeb\@extra@b@citeb \endcsname`.

3.3 `\include` `\cbinput` and `\cbunit`:

Redefine the `\include` macro so that when a file is `\include`'d … Increment the file number, (globally) update `\the@ipfilectr` to be `@-⟨number⟩`, and write `\gdef\@extra@binfo{@-⟨number⟩}` in the (main) `.aux` file, and do regular `\include`. When the `\include`'d file is finished, write `\gdef\@extra@binfo{}` in the (main) `.aux` file. Globally reset `\the@ipfilectr` to `{}`. Provide similar `\cbunit` and `\cbinput`.

3.4 `\cite`:

Redefine `\cite` (`\@citex` actually) and `\bibcite` to use the file number tag along with the specified tag. `\@citex` also uses `\citeform` as a hook to reformat each individual citation. Only do the redefinitions if no citation style that supports chapterbib has been loaded–as indicated by existence of filename tags.

3.5 `\nocite:`

A while ago LaTeX changed its `\nocite` so it checks the validity of the (no)citation tags. This necessitated adding the "`\@extra@b@citeb`" hook to `\nocite`. At first, I only redefined `\nocite` when redefining `\cite`, but there existed citation packages that supported `chapterbib` by putting `\@extra@b@citeb` into `\cite` but which did not define `\nocite` at all! These then gave warning messages for every `\nocite`. With version 1.7, I now try exercising the `\nocite` command with some functions disabled: (1) `.aux` file output is locally switched off; (2) The `\@ifundefined` command is hacked to merely typeset the text of the tag; (3) this is done in a disappearing `\hbox` so it generates no output. This detects (by setting a global flag `\@gtempa`) a `\nocite` command that executes `\@ifundefined` but not `\@extra@b@citeb`. In such cases, `\nocite` must be redefined.

3.6 `\citepunct` and `\citeform:`

These customization hooks are present in `cite.sty`; others (`\citemid`, `\citeleft`, `\citeright`) are not defined here because `\@cite` is not redefined. LaTeX has now adopted `\@cite@ofmt`, which is normally `\hbox`, so use both.

We redefine `\@testdef` so that it properly checks whether the `\cite` labels have changed.

3.7 Table of contents etc:

As of version 1.8, definitions of `\@extra@b@citeb` are written to the `toc`, `lot`, and `lof` files to avoid undefined citation tags in those tables. This is actually a problem in regular LaTeXwith BibTeX– cites that appear in those tables get numbered starting with 1, but they should be sequenced by where they originated – but the problem was worse with `chapterbib`.

3.8 `\CitationPrefix:`

This is a really ugly hack, but `\citeform` worked poorly for putting a chapter-identifier on each cite. It locally redefines both `\@bibitem` and `\@lbibitem` at each `\bibliography`. The insertion of the prefix is straightforward for `\@lbibitem`, but for `\@bibitem` the prefix is inserted by a redefined `\value` macro (eeek!).

4 Authors

Niel Kempson   (original)
Donald Arseneau `<asnd@triumf.ca>`   March 1993 – Mar 2009

5 Detailed List of Changes

Changes with ver 1.14: No warning from `\nocite*` in chapters, `\CitationPrefix`

Changes with ver 1.13: Rewrite final-bibs titling

Changes with ver 1.12: Update `\@citex` `\bfseries`

Changes with ver 1.11: Follow LaTeX's barring `\nocite` from preamble

Changes with ver 1.10: `\CBMainSectioning`; `\bibsection` hook

Changes with ver 1.9: `sectionbib` remove `\@mkboth`

Changes with ver 1.8: options `gather` and `duplicate`; `babel` workarounds; toc

Changes with ver 1.7: change the `\sectionbib` command re. headings.
update `\nocite` and alter `\citeform`

Changes with ver 1.6: change the `\sectionbib` command to take arguments.

Changes with ver 1.5: rootbib option; item 8 in instructions.

Changes with ver 1.4: `sectionbib` and `draft` options

Changes with ver 1.3: rewrite: The filename tag is defined in each `.aux` file by `\include`, not as an extra parameter to `\bibcite`. `\@bibitem` & `\@lbibitem` are left alone. Add `\citeform` and `\citepunct`. Add `\cbunit` and `\cbinput` for use without `\include`.