The **multibib** Package

Thorsten Hansen
Thorsten.Hansen@psychol.uni-giessen.de

v1.4 2008/12/10

Abstract

The multibib package allows to create references to multiple bibliographies within one document. It thus provides a complementary functionality to packages like bibunits or chapterbib, which allow to create one bibliography for multiple, but different parts of the document. The package introduces the generic macro `\newcites` which generates user-defined citation and bibliography commands for each new bibliography in the document. The new commands can be used intuitively, since both syntax and semantics are similar to their standard LaTeX equivalents. Each bibliography can have its individual style and BibTeX data file(s). Citations to each bibliography are collected in a new auxiliary file to be processed by BibTeX. The multibib package is compatible with cite, inlinebib, jurabib (to be loaded before multibib), natbib, suthesis and KOMA-Script classes.

1 Usage Notes

Suppose, you have to separate your citations into two bibliographies, one for primary literature and one for secondary literature. The multibib package defines the `\newcite` command which introduces a new family of cite and bibliography commands.

```
\newcites{sec}{Secondary Literature}
```

In this case the generic macro defines four new commands, suffixed by the first argument.

```
\citesec
\nocitesec
\bibliographystylesec
\bibliographysec
```

The second argument of `\newcites` provides the title of the corresponding bibliography. All these new commands accept the same parameters as their standard LaTeX equivalents and behave analogous to.—In the example above, the bibliography for primary literature can be generated by LaTeX's standard commands. When the document is processed, an auxiliary file `sec.aux` is generated, which needs to be compiled through BibTeX.

The tiny `\newcites` command is not limited to one bibliography. In fact, you can generate as much bibliographies as you like (only limited by the maximum number of TEX's output files, usually 16). In the following example, separate bibliographies are created for own articles, submitted articles, internal reports, and other work.

```
\newcites{own,submitted,internal}%
  {Own Work,%
   Submitted Work,%
   {Technical Reports, Master Theses and Ph.D. Theses}}
```

Note that a title containing commas needs to be grouped. Again, the bibliography for other work is generated by LATEX's standard commands.

`\newcites` After this introductory examples, let's look at the precise definition of `\newcites`. The command `\newcites{`⟨*suffix_list*⟩`}{`⟨*heading_list*⟩`}` defines a new family of commands for each suffix ⟨*s*⟩ in ⟨*suffix_list*⟩, namely `\cite`⟨*s*⟩, `\nocite`⟨*s*⟩, `\bibliographystyle`⟨*s*⟩, and `\bibliography`⟨*s*⟩ . The ⟨*suffix_list*⟩ is a comma separated list of letters, i.e, the same characters which are allowed as part of a TEX command. The ⟨*heading_list*⟩ is a comma separated list of words, i.e, the same sort of input (including commands) that can be used as argument of, e.g., `\section`. The commands `\cite`⟨*s*⟩ and `\nocite`⟨*s*⟩ generate citations that appear in the corresponding bibliography which is generated by `\bibliography`⟨*s*⟩. The title of this bibliography is defined by the entry ⟨*h*⟩ in ⟨*heading_list*⟩ at the same position as ⟨*s*⟩ in ⟨*suffix_list*⟩. The individual style of the bibliography can be defined by `\bibliographystyle`⟨*s*⟩.

`\setbiblabelwidth` When using numerical reference schemes like, e.g., plain, the labels in the bibliography have sometimes the wrong width. In this case you can set the desired label width using `\setbiblabelwidth{`⟨*number*⟩`}`, in the same fashion as you would specify the label width with `\thebibliography{`⟨*number*⟩`}`.

It is also possible to let LATEX do the job by using multibib's variant mbplain instead of plain as bibliography style. The modification does not affect the formatting style but only internal values, such that LATEX can determine the right width. More precisely, mbplain sets the longest label to the number of entries in the bibliography, while plain takes the smallest number which has the same width as the number of entries. For example, if your bibliography has 29 entries, plain sets the label to 10 while mbplain sets the label to 29. This is done by modifying the function `longest.label.pass`.

```
FUNCTION {longest.label.pass}
{ number.label int.to.str$ 'label :=
  number.label #1 + 'number.label :=
  label width$ longest.label.width >
  label width$ longest.label.width =   %% added in mbplain
  or                                   %% added in mbplain
    { label 'longest.label :=
      label width$ 'longest.label.width :=
    }
    'skip$
  if$
}
```

If you use other numerical styles than plain, you can customize your favorite style accordingly.

1.1 Package Options

labeled Add the internal suffixes for each bibliography as prefixes to the labels to the bibliography. For example, if you define

```
\newcites{A,B,C}%
   {Own \LaTeX\ Work,%
   Submitted,%
   {Habilitations, Dissertations and Diploma Thesis}}
```

and specify a numerical bibstyle `\bibliographystyleA{plain}`, the labels in the bibliography and the cited references will be [A1], [A2], [A3], . . . ; analogously, `\bibliographystyleB{plain}` results in labels and cited references [B1], [B2], [B3],. . . ; and `\bibliographystyleC{plain}` results in labels and cited references [C1], [C2], [C3], ... This option is useful together with option resetlabels and numerical reference schemes.

resetlabels When using numerical reference schemes, start each bibliography with '[1]'. Default is continuous numbering, such that labels are created unambiguously.

1.2 BibTeX Processing

For each bibliography a corresponding auxiliary file ⟨*s*⟩`.aux` is generated, that needs to be compiled through BibTeX. This can be done by the following bash-script.

```
#!/bin/bash
for file in *.aux ; do
  bibtex `basename $file .aux`
done
```

1.3 Limitations

- When numerical reference schemes are used and references to the same item appear in different bibliographies, the labels cannot be resolved properly. Rather, the replacement text (i.e, the number) is used which is defined in the auxiliary file read last.
- When using author-year reference schemes, entries with the same author and year combinations which appear in different bibliographies get the same label. In this case you have to manually add some hints to which bibliography the entry refers, e.g.,
  `\cite[in primary literature]{Foo:1999a}` vs.
  `\citesec[in secondary literature]{Foo:1999b}`.
  You can also edit the `.bbl` files created by BibTeX and add 'a', 'b',. . . to the labels, but your changes will be overwritten by subsequent BibTeX runs. The

differentiating letters ‘a’, ‘b’,… cannot be added automatically by BibTeX since the entries appear in different auxiliary files.

- With natbib, you cannot use numerical and author-year reference schemes together. Typographically, this is regarded bad style anyway.
- For jurabib and multibib, the order of loading is important: jurabib must be loaded before multibib.

2 Example

Let’s now consider a complete example with two bibliographies, one for TeX and LaTeX references and one for Postscript references. For demonstration purposes, different styles are used for the two references, namely alpha for TeX and LaTeX references and plain for Postscript references. Typographically, different bibliography styles are regarded as bad and therefore should be avoided in real documents. The reference data are supposed to be stored in a common BibTeX data file `lit.bib`. Note that the heading of the Postscript references is redefined with `\renewcommand{\refname}`.

```
\documentclass{article}
\usepackage{multibib}
\newcites{ltex}{\TeX\ and \LaTeX\ References}

\begin{document}
References to the \TeX book \citeltex{Knuth:1991}
and to Lamport’s \LaTeX\ book, which appears
only in the references\nociteltex{Lamport:1994}.
Finally a cite to a Postscript tutorial
\cite{Adobe:1985}.

\bibliographystyleltex{alpha}
\bibliographyltex{lit}

\renewcommand{\refname}{Postscript References}
\bibliographystyle{plain}
\bibliography{lit}

\end{document}
```

References to the TeXbook [Knu91] and to Lamport’s LaTeX book, which appears only in the references. Finally a cite to a Postscript tutorial [1].

TeX and LaTeX References

[Knu91] Donald E. Knuth. *The TeX book.* Addison-Wesley, Reading, Massachusetts, 1991.

[Lam94] Leslie Lamport. *LaTeX: A Document Preparation System.* Addison-Wesley, Reading, Massachusetts, 2 edition, 1994.

Postscript References

[1] Adobe System Incorporated. *Postscript Language Tutorial and Cookbook.* Addison-Wesley, Reading, Massachusetts, 1985.

Figure 1: Example input and output.

To process your document, three runs of LaTeX and two runs of BibTeX are required.

```
latex mydoc
bibtex mydoc
bibtex ltex
latex mydoc
latex mydoc
```

3 Notes for Class and Package Writers

Packages such as cite or natbib which modify the internal macro `\@citex` called from `\cite` by, e.g., changing the number of arguments, and/or define new variants of `\cite`, such as natbib's `\citep`, deserve a special handling to ensure compatibility to multibib. In the first case, the problem arise because multibib's redefinition of `\@citex` to write to a new auxiliary file `\@newciteauxhandle` instead of writing to `\@auxout` is overwritten. In the second case, the problem arise because multibib is unaware of additional cite commands and thus cannot define the new variants. The problems can be solved by using the compatibility mechanisms supported by multibib as described in the following.

3.1 Packages which modify `\@citex`

Packages which modify `\@citex` can be made compatible to multibib by making the following three changes to the code:

1. Add the following code:

   ```
   \providecommand\@newciteauxhandle{\@auxout}
   \def\@restore@auxhandle{\gdef\@newciteauxhandle{\@auxout}}
   \AtBeginDocument{%
     \@ifundefined{newcites}{\global\let\@restore@auxhandle\relax}{}}
   ```

   The first line defines `\@newciteauxhandle` to the default value `\@auxout` if `\@newciteauxhandle` is not already defined. The second line defines macro `\@restoreauxhandle` which sets `\@newciteauxhandle` to its default value. The next two lines `\let` macro `\@restoreauxhandle` to `\relax` in case multibib is not loaded and `\newcites` is undefined.

2. In the redefinitions of `\@citex` replace `\@auxout` by `\@newciteauxhandle`.

3. At the end of the each `\@citex` redefinitions add the macro `\@restoreauxhandle`.

At present, packages cite and jurabib use this compatibility mechanism.

3.2 Packages which define new cite commands

Packages which define new cite commands can add these commands using `\@mb@citenamelist`. The default definition, which already includes natbib's cite variants, is given below.

```
\def\@mb@citenamelist{cite,citep,citet,citealp,citealt}
```

A package which defines, e.g, a new cite command `\footcite` informs multibib about this command by defining

```
\def\@mb@citenamelist{cite,citep,citet,citealp,citealt,footcite}
```

At present, package jurabib use this compatibility mechanism.

3.3 Advantages of the compatibility mechanism

Compared to a solution where all compatibility patching is done by multibib, such as is presently the case for natbib, the above solution has several advantages:

1. The mechanisms provides a general solution for several packages. Any new package can be made compatible with multibib without the need for a new release of multibib.
2. Existing packages can change the number of arguments to `\@citex` or introduce new cite macros without the need to change multibib.
3. If `\@citex` needs to read the next input character, this character is available and not masked by multibib's `let`s which follow the `\@std@citex`. This is important for the cite package.
4. More efficient code. A single `\let` in `\@restore@auxhandle` instead of four.

4 Brief Discussion of Related Packages

Similar packages. The bibtopic package addresses the same issue as the present package, using a completely different approach. The user has to provide a separate BibTeX data file `liti.bib` for each bibliography and can generate the corresponding bibliography within a `btSect` environment.

```
\begin{btSect}{liti}
  \btPrintCited
\end{btSect}
```

Using `\btPrintNotCited`, a complementary bibliography showing all entries not cited in the document can also be generated. Additionally, an environment `btunit` is defined to generate bibliographies for different units of the text, thus partly offering the functionality of packages like bibunits or chapterbib. Besides its powerful functionality, the bibtopic package has some drawbacks and restrictions:

- less intuitive user interface
- the user has to split his BibTeX data files
- bibliography styles that put some material between the begin of `thebibliography` and the first `\bibitem` deserve a special handling
- does not work with bibliography style `unsorted`
- and more, see Sec. 'Restrictions' of the bibtopic documentation

Finally, due to its large size (1059 lines) and implementation approach, processing is probably slower. In a nutshell, bibtopic does the following: for each bibliography, a `\citation{*}` command is written to a new, corresponding auxiliary file, such that *all* entries of the BibTeX data base are extracted. Then, by reimplementing the `\bibitem` command, all items which do not occur in a special list of labels, produced by the cite commands, are deleted. Obviously, doing part of BibTeX's work requires some amount of memory and time (and programming skills, noteworthy `;-)`).

Complementary packages. The packages bibunits and chapterbib provide commands to generate multiple bibliographies, too. The main difference is, that with these packages a single bibliography is generated for multiple, but different parts of the text. Therefore, within a certain unit, i.e., part of the text, citations can only be made to a single, corresponding bibliography. On the other hand, with multibib, within the same (but single) part, citations can be generated for multiple bibliographies.

Analogous packages. The multind package allows multiple indexes.

5 Acknowledgments

The author thanks Donald Arseneau for suggesting the compatibility mechanism for packages which modify `\@citex`. Further, the author acknowledges the contributions of numerous people, in particular Jens Berger and Frank Mittelbach, whose suggestions and bug reports have helped to improve multibib.

6 The Macros

One paradigm which guided the development of the code was to use standard LaTeX commands as much as possible, and to customize their behavior using `\let`.

```
⟨*package⟩
```

6.1 Option Handling

`\ifcontinuouslabels`

Define a new if to switch between continuous labeling of bibliographies (default) and start of labels with '[1]'. The latter can be activated with option resetlabels.

```
\newif\ifcontinuouslabels
\continuouslabelstrue
\DeclareOption{resetlabels}{\continuouslabelsfalse}
```

`labeled`

Define a new option to add the internal suffixes for each bibliography as labels to the bibitems.

```
\newif\iflabeled
\labeledfalse
\DeclareOption{labeled}{\labeledtrue}
```

Finally, process all package options.

```
\ProcessOptions
```

6.2 Preliminaries

`\mylop`
`\mylopoff`

Variation of `\lop`, using '`,`' as separator instead of '`\`' to extract the elements in the first argument of `\newcites` (cf. The TeXbook, p. 378).

```
\def\mylop#1\to#2{\expandafter\mylopoff#1\mylopoff#1#2}
\long\def\mylopoff#1,#2\mylopoff#3#4{\def#4{#1}\def#3{#2}}
```

`\@newciteauxhandle`

Define a new handle of the auxiliary file for all `\cite` and `\nocite` commands, standard as well as newly defined. For the standard commands, this handle is let to `\auxout`. Since version 1.1b the definition `\let\@newciteauxhandle\@auxout` is no longer needed since below we reset `\@citex` to its standard value after each call of `\mb@@citex`.

`\std@@citex`
`\mb@@citex`

Definitions of `\mb@@citex` to replace `\@citex` in the definition of the various `cite` commands. The idea is to write to a specific file referred to `\@newciteauxhandle` instead of `\@auxout`. Below in the definition of the specific `\cite<s>` and `\nocite<s>` commands, `\@newciteauxhandle` is set to the new auxiliary files `<s>.aux`. If `\@newciteauxhandle` is already defined by compatible packages, simply `\let` macro `\mb@@citex` to `\@citex`

If `natbib` is loaded, a variant of `\@citex` with an additional optional argument is needed. Macro `\NAT@set@cites` invokes `natbib`'s `\cite` and `\@citex` definitions.

```
\AtBeginDocument{%
  \@ifpackageloaded{natbib}{%
    \NAT@set@cites
    \let\std@@citex\@citex
    \def\mb@@citex[#1][#2]#3{%
      \global\let\@save@auxout\@auxout
      \let\@auxout\@newciteauxhandle
      \std@@citex[#1][#2]{#3}%
      \let\@auxout\@save@auxout
      \let\@citex\std@@citex}%
  }{% else
    \@ifundefined{@newciteauxhandle}{% not defined by compatible packages
      \let\std@@citex\@citex
      \def\mb@@citex[#1]#2{{%
        \let\@auxout\@newciteauxhandle
        \std@@citex[#1]{#2}}%
        \let\@citex\std@@citex}%
      }{\let\mb@@citex\@citex}%
  }%
}%
```

`\@newusecounter`
`\newusecounter`

Do not reset counter to zero at begin of bibliography. This generates continuous numbering of references through all new bibliographies to ensure that numerical references are created unambiguously. To reset the counter only for the *first* bibliography, `\newusecounter` is initially let to `\usecounter`. Below, in the definition of the new bibliographies, `\newusecounter` is let to `\@newusecounter`.

```
\def\@newusecounter#1{\@nmbrlisttrue\def\@listctr{#1}}
\let\newusecounter\usecounter
```

`\std@bibliography`
`\bibliography`

To ensure continuous numbering, the `\newusecounter` is redefined also for the standard `\bibliography`, same as for the new `\bibliography`$\langle s\rangle$ macros.

```
\let\std@bibliography\bibliography
\def\bibliography#1{%
  \ifcontinuouslabels
    \let\usecounter\newusecounter
  \fi
  \std@bibliography{#1}%
  \ifcontinuouslabels
```

```
   \global\let\newusecounter\@newusecounter
 \fi}
```

`\mb@biblabelwidth`
`\setbiblabelwidth`

The macro `\setbiblabelwidth` explicitly sets the width of the labels in the bibliography. An internal counter `\mb@biblabelwidth` is introduced and a macro `\setbiblabelwidth` which simply assigns its argument to the new counter. The value of the counter is then used in multibib's modification of `\thebibliography`.

```
\newcount\mb@biblabelwidth
\newcommand\setbiblabelwidth[1]{\mb@biblabelwidth #1}
```

`\std@thebibliography`
`\thebibliography`

The argument of `\thebibliography` determines the width of the labels in the bibliography. First it is checked if the parameter is a number. If it is not a number, we do not have to care about setting label widths and can simply call `\std@thebibliography`. Otherwise we check if the label width is not set explicitly, i.e., `\mb@biblabelwidth=0`. In this case the label width is determined as the the last label in the previous bibliography `\c@enumiv` plus the number of labels in the current bibliography `#1` (provided the bibliography style computes the values right, like mbplain). If the value is set explicitly by `\mb@biblabelwidth`, this value is used and then reset to zero such that only the labels in the current bibliography are affected.

```
\AtBeginDocument{%
 \@ifpackageloaded{suthesis}%
   {\def\thebibliography#1{%
      \newpage
      \addcontentsline{toc}{chapter}{\bibname}%
      \@ldthebibliography{#1}}}%
   {}%
 \@ifpackageloaded{natbib}%
   {%


\renewenvironment{thebibliography}[1]{%
\bibsection\parindent \z@\bibpreamble\bibfont\list
  {\@biblabel{\arabic{NAT@ctr}}}{\@bibsetup{#1}%
   \usecounter{NAT@ctr}}% %% only changed here to usecounter
   \ifNAT@openbib
     \renewcommand\newblock{\par}
   \else
     \renewcommand\newblock{\hskip .11em \@plus.33em \@minus.07em}%
   \fi
   \sloppy\clubpenalty4000\widowpenalty4000
   \sfcode`\.=1000\relax
   \let\citeN\cite \let\shortcite\cite
   \let\citeasnoun\cite
}{\def\@noitemerr{%
 \PackageWarning{natbib}
    {Empty `thebibliography' environment}}%
 \endlist\vskip-\lastskip}

     \let\std@thebibliography\thebibliography
     \def\thebibliography#1{%
       \@isnumber{#1}%
       {\ifnum\mb@biblabelwidth=0
```

```
          \@tempcnta\c@NAT@ctr %% changed here to c@NAT@ctr
          \ifcontinuouslabels
          \advance\@tempcnta#1%
          \fi
          \std@thebibliography{\@arabic\@tempcnta}%
          \else
          \std@thebibliography{\@arabic\mb@biblabelwidth}%
        \global\mb@biblabelwidth 0
        \fi}%
        {\std@thebibliography{#1}}%
     }%
   }%
   {% else, natbib not loaded
     \let\std@thebibliography\thebibliography

     \def\thebibliography#1{%
        \@isnumber{#1}%
          {\ifnum\mb@biblabelwidth=0
            \@tempcnta\c@enumiv
            \ifcontinuouslabels
            \advance\@tempcnta#1%
            \fi
            \std@thebibliography{\@arabic\@tempcnta}%
            \else
            \std@thebibliography{\@arabic\mb@biblabelwidth}%
          \global\mb@biblabelwidth 0
          \fi}%
        {\std@thebibliography{#1}}%
    }%
   }%
}
```

`\@isnumber` Macro `\@isnumber` behaves as follows: If `#1` is a number (which may contain blanks at arbitrary positions), `#2` is executed, else `#3`.

```
\newcommand{\@isnumber}[3]{%
  \def\argdef{#1}%
  \edef\argedef{#1}%
  \ifx\argedef\empty
    #3
  \else
    \ifx\argdef\argedef
      \global\@tempswafalse
      \setbox\@tempboxa=\hbox{\@scannumber#1\plugh}%
      \if@tempswa#2\else#3\fi
    \else
      #3
    \fi
  \fi}
```

`\@scannumber` Macro `\@scannumber` evaluates the single characters of its argument. If all characters are digits or blanks,`\@scannumber` is recursively evoked until it finally reads `\plugh` which marks the end of the argument (in macro `\@isnumber`) and sets `\@tempswatrue` which is then used in `\@isnumber`. If the tested character is not a digit, the macro exits, leaving `\@tempswa` unchanged.

```
\let\plugh\relax
\newcommand{\@scannumber}[1]{%
  \let\testchar#1%
  \ifx \testchar 0\let\next\@scannumber
  \else\ifx \testchar 1\let\next\@scannumber
  \else\ifx \testchar 2\let\next\@scannumber
  \else\ifx \testchar 3\let\next\@scannumber
  \else\ifx \testchar 4\let\next\@scannumber
  \else\ifx \testchar 5\let\next\@scannumber
  \else\ifx \testchar 6\let\next\@scannumber
  \else\ifx \testchar 7\let\next\@scannumber
  \else\ifx \testchar 8\let\next\@scannumber
  \else\ifx \testchar 9\let\next\@scannumber
  \else\ifx \testchar \plugh \let\next\relax \global\@tempswatrue
  \else \let\next\relax
  \fi\fi\fi\fi\fi\fi\fi\fi\fi\fi\fi
  \next}
```

`\@gobbletillplugh` Macro `\@gobbletillplug` gobbles all input up to (and including) `\plugh` by defining `\def\@gobbletillplugh#1\plugh{}`. Since version 1.1d, `\@scannumber` is evaluated in a `\hbox` and the gobbling is no longer needed.

`\mb@addtocontents` Some styles like suthesis or KOMA-Script classes like scrartcl with option bibtotoc redefine `\bibliography` such that the heading of the bibliography appears in the table of contents. Entries for the table of contents are generated by writing the appropriate information to `\@auxout`. Because `\@auxout` is locally redefined when the bibliography is read, entries go to the wrong file. Therefore, a macro `\mb@addtocontents` is introduced which writes to `\temp@auxout`. Below, in the definition of the new `\bibliography`⟨*s*⟩, `\mb@addtocontents` replaces the standard `\addtocontents`, and `\temp@auxout` is set to the default `\@auxout`.

```
\long\def\mb@addtocontents#1#2{%
  \protected@write\temp@auxout
    {\let\label\@gobble \let\index\@gobble \let\glossary\@gobble}%
    {\string\@writefile{#1}{#2}}}
```

`\bibname` Package multibib redefines the headings of the new bibliographies using `\refname`. Some classes like book and report use `\bibname` instead of `\refname`. Instead of changing the definition of `\bibname`, we simply define `\refname` in the same was as `\bibname` in the `newcites` loop below.

`\@mb@citenamelist` Define list of cite commands to be processed within the `\newcites` loop below. If already defined by other packages, `\relax`.

```
\@ifundefined{@mb@citenamelist}%
  {\def\@mb@citenamelist{cite,citep,citet,citealp,citealt}}%
  {\relax}
```

6.3 The `newcites` loop

`\newcites` Loop over all headings in ⟨*heading_list*⟩. The current heading is stored in `\@newrefname`, the corresponding suffix is stored in `\@suffix`.

```
\def\newcites#1#2{%
```

```
\def\@suffixlist{#1,}%
\@for\@newrefname:=#2\do{%
  \mylop\@suffixlist\to\@suffix
```

`\@refname<s>` Define the reference title `\protected`, such that the title can contain control sequences as, e.g., in the title 'LaTeX References'.

```
  \expandafter\protected@edef\csname refname\@suffix\endcsname
    {\@newrefname}%
```

`\@auxout<s>`
`\@auxout<s>name` Define new write. Input the auxiliary file if it exists before opening it, to read the replacement text for the `\cite` commands which is generated by `\bibcite`. Because some styles (like inlinebib.sty) write command names containing an `@` to the auxiliary files, `\makeatletter` is locally set.

```
  \if@filesw
    \expandafter\newwrite\csname @auxout\@suffix\endcsname
    \expandafter\edef\csname @auxout\@suffix name\endcsname{\@suffix}%
    \begingroup
      \makeatletter
      \@input@{\csname @auxout\@suffix name\endcsname .aux}%
    \endgroup
    \immediate\openout\csname @auxout\@suffix\endcsname
                      \csname @auxout\@suffix name\endcsname .aux
  \fi
```

Define new `\cite` and `\nocite`. For `\nocite`, we simple let `\@auxout` to the new auxiliary file. For `\cite`, this approach does not work because `\cite` can have an optional argument, so we cannot enclose the redefinition of `\@auxout` in braces. We slightly modify `\@citex` (see above) to use a special handle, `\@newciteauxhandle`, which is let to the new auxiliary file.

`\cite<s>` For each ⟨*citename*⟩ in `\@mb@citenamelist`, define multibib's new ⟨*citename*⟩⟨*s*⟩ commands. For example, if ⟨*citename*⟩ := `cite` and if ⟨*s*⟩ := `own`, a command `\citeown` is defined as:

```
\def\citeown{%
  \let\@citex\mb@@citex
  \let\@newciteauxhandle\@auxoutown
  \cite}
```

```
  \@for\@citename:=\@mb@citenamelist\do{%
    \expandafter\edef\csname \@citename\@suffix\endcsname{%
      \let\noexpand\@citex\noexpand\mb@@citex
      \let\noexpand\@newciteauxhandle\csname @auxout\@suffix\endcsname
      \noexpand\csname\@citename\endcsname}%
  }%
%
```

`\nocite<s>` Define new `\nocite`⟨*s*⟩, e.g., if ⟨*s*⟩ := `own`, a command `\nociteown` is defined as:

```
\def\nociteown##1{{%
  \let\@auxout\@auxoutown
  \nocite{##1}}}
```

```
\expandafter\edef\csname nocite\@suffix\endcsname##1{{%
  \let\noexpand\@auxout\csname @auxout\@suffix\endcsname
  \noexpand\nocite{##1}}}%
```

`\bibitem<s>`

```
\iflabeled % if option labeled
  \expandafter\edef\csname @bibitem\@suffix\endcsname##1{%
    \noexpand\item
    \noexpand\if@filesw \noexpand\immediate\noexpand\write\noexpand\@auxout
    {\noexpand\string\noexpand\bibcite{##1}%
       {\@suffix\noexpand\the\noexpand\value{\noexpand\@listctr}}}%
    \noexpand\fi
    \noexpand\ignorespaces}%
  \expandafter\edef\csname @biblabel\@suffix\endcsname##1{[\@suffix##1]}%
\fi % end if option labeled
```

`\bibliography<s>` Define new `\bibliography`⟨*s*⟩ equivalents. The standard `\bibliography` macro does two things: The bibdata file is written to the auxiliary file `\@auxout` and the `.bbl` file `\jobname.bbl` is inputed. For `\bibliography`⟨*s*⟩, we thus let both `\@auxout` and `\jobname` refer to the new auxiliary file ⟨*s*⟩`.aux`. Since some styles and classes write the heading of the bibliography to the table of contents, the old meaning of `\@auxout` is saved in `\temp@auxout` and `\addtocontents` is locally replaced by multibibs variant `\mb@addtocontents` (defined above) which writes to `\temp@auxout` instead of `\auxout`. Further, the heading of the bibliography generated by `\refname` or `\bibname` depending on the class is set to the particular heading of `\bibliography`⟨*s*⟩. After this redefinitions, the standard `\bibliography` can be evoked. Extra braces are needed to encapsulate the various `\let`s.

```
\expandafter\edef\csname bibliography\@suffix\endcsname##1{{%
  \let\noexpand\temp@auxout\noexpand\@auxout
  \let\noexpand\addtocontents\noexpand\mb@addtocontents
  \let\noexpand\@auxout\csname @auxout\@suffix\endcsname
  \let\noexpand\jobname
     \expandafter\noexpand\csname @auxout\@suffix name\endcsname
  \let\noexpand\refname
      \expandafter\noexpand\csname refname\@suffix\endcsname
  \let\noexpand\bibname
      \expandafter\noexpand\csname refname\@suffix\endcsname
  \iflabeled
    \let\noexpand\@bibitem
        \expandafter\noexpand\csname @bibitem\@suffix\endcsname
    \let\noexpand\@biblabel
        \expandafter\noexpand\csname @biblabel\@suffix\endcsname
  \fi
  \noexpand\bibliography{##1}%
  }}
```

`\bibliographystyle<s>` Define new `\bibliographystyle`⟨*s*⟩ equivalents.

```
200     \expandafter\edef\csname bibliographystyle\@suffix\endcsname##1{%
201       \noexpand\if@filesw
202         \noexpand\immediate\noexpand\write\csname @auxout\@suffix\endcsname
203           {\noexpand\string\noexpand\bibstyle{##1}}%
204       \noexpand\fi}
```

Close loop and end macro.

```
205   }%
206 }
```

Finally, restrict the use of `\newcites` to the preamble such that the auxiliary files can be `\@input@`ed.

```
207 \@onlypreamble\newcites
208 ⟨/package⟩
```

Index

Numbers written in italic refer to the page where the corresponding entry is described; numbers underlined refer to the code line of the definition; numbers in roman refer to the code lines where the entry is used.

Change History

Version numbers suffixed with characters have not been released.

v1.0
General: Initial version 1

v1.0a
`\bibliography`: Continuous numbering also for `\bibliography`. 8

v1.0b
`\thebibliography`: Correct label width for numerical reference schemes. 8

v1.0c
`\@auxout<s>name`: `\makeatletter` when input new auxiliary files. 11

v1.0d
`\mb@addtocontents`: Add bibliography name to table of contents. 11

v1.0e
- `\setbiblabelwidth`: New macro. . . 8
- `\thebibliography`: Works with `\setbiblabelwidth`. 8

v1.0f
- `\mb@@citex`: Update to natbib v7.0: save `\@auxout` before redefinition instead of encapsulation in braces. 7

v1.0g
- `\@isnumber`: Empty string is not handled as number. 10

v1.0h
- `\bibname`: Changed to evoke `\refname` instead. 11

v1.0i
- General: New option resetlabels. . . 7
- `\bibliography`: Change to respond to option resetlabels. 8
- `\bibliography<s>`: Remove setting of `\usecounter` and `\newusecounter` which is done in redefined `\bibliography`. . 12
- `\thebibliography`: No change of label width for option resetlabels. 8

v1.0j
- `\bibname`: New implementation: Check if `\bibname` is undefined instead of checking if class book or report is loaded. 11

v1.0k
- `\thebibliography`: suthesis compatibility. 8

v1.1
- General: Documentation. 1

v1.1a
- General: Equivalents of macros `\citep` and `\citet` added. 1

v1.1b
- General: Evoke multibib's `\@citex` variant not for standard `\cite` commands. 1
- `\@newciteauxhandle`: Handle removed. 7
- `\mb@@citex`: `\global` setting of `\@save@auxout`; setting of `\newciteauxhandle` removed; instead, reset `\citex` to its standard version. 7

v1.1c
- General: Equivalents of macros `\citealp` and `\citealt` added. 1

v1.1d
- General: Spurious letters occurring for bibstyle alpha removed. . . . 1
- `\@gobbletillplugh`: Macro removed. 10
- `\@isnumber`: Execute `\@scannumber` in `\hbox`. 10
- `\@scannumber`: Initialize `\plugh` with `\relax`; simply `\relax` when the `\testchar` is not a digit instead of `\@gobbletillplugh`. 10

v1.1e
- General: Additional definition of `\refname` in `\newcites` loop. . . 1
- `\bibliography<s>`: Definition of `\bibname` analogously to `\refname`. 12
- `\bibname`: Macro removed. 11

v1.2
- General: Documentation. 1

v1.2a
- General: General compatibility mechanism for packages which redefine `\@citex`, motivated by compatibility to cite v4.0. 1
- `\mb@@citex`: New definition of `\mb@@citex` only if `\@newciteauxhandle` is not already defined by compatible packages. 7

v1.2b
- General: General compatibility mechanism for packages which define their own cite commands, motivated by compatibility to jurabib. 1
- `\@mb@citenamelist`: Macro added. 11
- `\cite<s>`: Loop over `\@mb@citenamelist` added. 12

v1.2c
- General: Code streamlined. 1
- `\nocite<s>`: `\let` `\@citex` `\mb@@citex` removed. 12

v1.3
- General: Documentation. 1

v1.3a
- General: Continous numerical labels with natbib. 1
- `\thebibliography`: Natbib specific definition of thebibliography, modified to use `\usecounter`. . 8

v1.3b

- General: `\isnumber` changed to accept arguments with `\fi`, e.g., `\iffalse`bjko`\fi`. 1
- `\@isnumber`: `\isnumber` changed to accept arguments with `\fi`, e.g., `\iffalse`bjko`\fi`. 10

v1.3c

- General: Allow `\bibliographystyle<s>` to be used in the preamble. . . . 1
- `\bibliographystyle<s>`: Write directly the `\bibstyle` macro to the specific auxfile instead of redefining a auxhandle and then using the standard `\bibliographystyle` macro. . 14

v1.3d

- General: New option labeled. 1
- `labeled`: Define option labeled. . . . 7
- `\bibitem<s>`: Define a bibliography specific version of `\@bibitem` and `\@biblabel` if labeled is active. 12
- `\bibliography<s>`: Activate the bibliography specific versions of `\@bibitem` and `\@biblabel` if labeled is active. 12

v1.4

- General: Documentation. 1