

The bibunits package

Thorsten Hansen
hansen@neuro.informatik.uni-ulm.de

v2.0 1999/05/22

Abstract

The bibunits package allows to generate separate bibliographies for different units (parts) of the text. The units can be chapters, sections or bibunit-environments. The style will separate the citations of each unit of text into a separate file to be processed by `BIBTEX`. The same cited item can occur in more than one bibliography. A global bibliography can also appear in the document and citations can be placed in both at the same time. The bibunits package ist based on the bibunits style by Jose Alberto Fernandez (alberto@cs.umd.edu).

1 Usage notes

<code>\bibliographyunit</code>	Use <code>\bibliographyunit[⟨unit⟩]</code> , where <code>⟨unit⟩</code> can be <code>\chapter</code> or <code>\section</code> to specify for which document unit references must be generated, namely for every chapter or for every section. Use <code>\bibliographyunit</code> with no arguments to deactivate bibliographyunits. By default <code>\bibliographyunit</code> is deactivated.
<code>\bibliography</code> <code>\bibliographystyle</code>	You can create a global bibliography as usual with the commands <code>\bibliography[⟨BibTeX files⟩]</code> and <code>\bibliographystyle[⟨style⟩]</code> . If <code>\bibliographyunit</code> is active, these commands also specify the <code>BIBTEX</code> files and style to be used by default in the local units.
<code>\bibliography*</code> <code>\bibliographystyle*</code>	When <code>\bibliographyunit</code> is active, you can use starred forms <code>\bibliography*{⟨BibTeX files⟩}</code> and <code>\bibliographystyle*{⟨style⟩}</code> to specify the defaults for the local units only. These commands do not generate any information for the global bibliography.
<code>bibunit</code>	The environment <code>\begin{bibunit}[⟨style⟩]</code> allows the creation of a unit while <code>\bibliographyunit</code> is not active. The optional parameter <code>⟨style⟩</code> specifies a style for the bibliography different from the default, if any. Warning: The use of this environment while <code>\bibliographyunit</code> is active can produce strange results. Note that default style and <code>BIBTEX</code> files only exist if specified while <code>\bibliographyunit</code> is active.
<code>\cite*</code> <code>\nocite*</code>	Use <code>\cite</code> and <code>\nocite</code> to generate citations that appear in the local bibliography. Use <code>\cite*</code> and <code>\nocite*</code> inside a unit to generate citations for both the local and global bibliography.

`\putbib` You must insert the command `\putbib[BibTeX files]` before the end of each unit at the location where you want the bibliography to be inserted. If the optional argument is omitted, `\putbib` uses the default `BIBTEX` files, if any.

1.1 `BIBTEX` processing

For each bibunit, in sequence, there is now a corresponding file `bui.aux` that need to be compiled through `BIBTEX`. Suppose your document has *n* different bibunits, you must now invoke `BIBTEX` on `bu1, ..., bun`. This can be done by a `csh`-script.

```
#!/bin/csh
foreach auxfile (bu*.aux)
  echo bibtex 'basename $auxfile .aux'
  bibtex 'basename $auxfile .aux'
end
```

If by some strange coincidence you have named some of your files `bui.aux`, you have to redefine the internal command `\@bibunitname`, otherwise your files will be overwritten. To get for your document `foo.tex` filenames `foo.i.aux` as for the first version of bibunits, you can redefine the internal macro `\@bibunitname`.

```
\makeatletter
\renewcommand{\@bibunitname}{\jobname.\the\@bibunitauxcnt}
\makeatother
```

If you also want a global bibliography for your document `foo.tex`, the file `foo.aux` needs to be compiled through `BIBTEX` as well.

2 Examples

2.1 Bibunits by the `bibunit` environment

In the first example, two bibliographies are generated, the first using the `BIBTEX` file `texlit.bib` and the style `plain`, the second using the `BIBTEX` file `lit.bib` and the style `abbrv`.

```
\documentclass{article}
\usepackage{bibunits}

\begin{document}
\begin{bibunit}[plain]
  some text \cite{lamport:1994} more text more citations
  \putbib[texlit]
\end{bibunit}
some text between the units
\begin{bibunit}[abbrv]
```

```

    some text \cite{gnu:1998} more text more citations
    \putbib[lit]
\end{bibunit}
\end{document}

```

If all bibunits use the same BIB_TE_X files and style, you can specify defaults and omit the optional arguments of the `bibunit` environment and the `\putbib` macro. In the second example, a default BIB_TE_X file `texlit.bib` and a default style `abbrv` is defined. Note that you have to activate bibliographyunits with the command `\bibliographyunit[\section]` (`\chapter` would also work) before you can specify the defaults, and afterwards you have to deactivate bibliographyunits with the command `\bibliographyunit` before you can use the `bibunit` environment.

```

\begin{document}
\bibliographyunit[\section]
\bibliography*{texlit}
\bibliographystyle*{plain}
\bibliographyunit

\begin{bibunit}
  some text \cite{lamport:1994} more text more citations
  \putbib
\end{bibunit}
some text between the units
\begin{bibunit}
  some text \cite{knuth:1991} more text more citations
  \putbib
\end{bibunit}
\end{document}

```

If you use the class `article`, the heading of the bibliography has the same size as a section, which might be too large, especially if you have bibunits *within* sections or even subsections. In this case you may want to change the bibliography heading to have the same appearance as a subsection.

```

\let\stdthebibliography\thebibliography
\renewcommand{\thebibliography}{%
  \let\section\subsection
  \stdthebibliography}

```

Note that other classes may use other formatting instead of `section` for the bibliography heading. In this case the definition has to be changed accordingly, e.g., for the `book` class, where bibliography headings appear at the chapter level, let `\chapter` to `\subsection`.

```

\let\stdthebibliography\thebibliography
\renewcommand{\thebibliography}{%
  \let\chapter\subsection
  \stdthebibliography}

```

2.2 Bibunits by chapters or sections

You can also define bibliographies for every chapter or section. In this case, the redefinition of `\thebibliography` is essential for the proper behavior of the `bibunits` package. The reason is quite simple: if you create bibliographies for every section, and `\thebibliography` also appears at the section level, a new unit is open by `\thebibliography`, and the information of the previous unit is no longer available. (More technically: The auxiliary file of the previous unit is closed, and the replacement text for the citation which is generated inside `\thebibliography` cannot be written to this file. Instead, it is written to the global `.aux` file.)

The next example corresponds to the first example of the previous section with different `BIBTEX` files and styles. Note that you have to specify the `\bibliographystyle*` *before* the corresponding section.

```

\documentclass{article}
\usepackage{bibunits}

\let\stdthebibliography\thebibliography
\renewcommand{\thebibliography}{%
  \let\section\subsection
  \stdthebibliography}

\begin{document}
\bibliographyunit[\section]

\bibliographystyle*{plain}
\section{First section}
  some text \cite{lampport:1994} more text more citations
  \putbib[texlit]
some text between the units
\bibliographystyle*{abbrv}
\section{Second and last section}
  some text \cite{gnu:1998} more text more citations
  \putbib[lit]
\end{document}

```

You can also use the same `BIBTEX` file and style for all units.

```

: same as in the previous example

\begin{document}

```

```

\bibliographyunit[\section]
\bibliography*{texlit}
\bibliographystyle*{plain}

\section{First section}
  some text \cite{lampport:1994} more text more citations
  \putbib
some text between the units
\section{Second and last section}
  some text \cite{knuth:1991} more text more citations
  \putbib
\end{document}

```

In all four examples, one can specify a global bibliography and its style with the usual \LaTeX commands. Citations for the global bibliography are entered using `\cite` and `\nocite` commands while outside a unit or using `\cite*` and `\nocite*` while inside a unit. The starred forms generate citations also for the local unit. Thus it is not possible to generate citations *only* for the global bibliography while inside a unit (which seems to be a reasonable restriction). Note that if `\bibliographyunits` is active, *and* you use a global bibliography, *and* you use the same \BIBTeX file and style for all (local as well as global) bibliographies, *then* the `\bibliography*` and `\bibliographystyle*` commands are not necessary. In this case proper defaults for the local bibliographies are specified by the commands `\bibliography` and `\bibliographystyle`, too. Note that a second run of \LaTeX is necessary before \BIBTeX processing, in case that the global bibliography has its usual place at the end of the document.

3 Changes

The following features have been added from version 1.0 to 2.0:

- The same referenced item can be used in more than one bibliography.
- The package can be used on systems where filenames of three items are not allowed.

Instead of creating files `\jobname.<i>.aux`, the files are named `bu<i>.aux`. The old filename of three items causes problems on none-UNIX-systems like MS-DOS, CMS or VMS, as reported, e.g., in the \LaTeX Companion.

- Auxiliary files are created only for bibunits which contain cite commands.

Imagine, you want to specify bibliographies for sections using `\bibliographyunit[\section]`, but only a small number of, say, ten sections of your total of 100 sections have cite commands. Formerly, 100 auxiliary files have been created by `bibunits`, only ten. This is faster and makes both manual and automatic processing by \BIBTeX easier.

- The default styles and files are defined globally, so that the information is already present at the first run of \LaTeX .

Formerly, default styles and files have been written *only* to the global auxiliary file, so that a second run of \LaTeX was necessary before \BibTeX processing.

These changes along with the upgrading of the package to $\text{\LaTeX}2_{\epsilon}$ have been done by Thorsten Hansen. The primary coding of `bibunits` has been done by Jose Alberto Fernandez.

4 The Macros

```
1 \langle *package \rangle
```

4.1 An auxiliary file for each bibunit is provided

<code>\@bibunitaux</code>	Define the file descriptor <code>\@bibunitaux</code> of the auxiliary file that is generated for each bibunit.
	<code>2 \newwrite\@bibunitaux</code>
<code>\@bibunitauxcnt</code>	Define the counter for the bibunits and initialize it with zero.
	<code>3 \newcount\@bibunitauxcnt \@bibunitauxcnt=0</code>
<code>\@bibunitname</code>	The command <code>\@bibunitname</code> sets the basename of the auxiliary files that are created for each bibunit to <code>bu</code> (bu for bibunits), followed by the current number of the bibunit, <code>\the\@bibunitauxcnt</code> . (In version 1.0 of <code>bibunits</code> there was no such command, and the command sequence <code>\jobname.\the\@bibunitauxcnt</code> was directly used.)
	<code>4 \def\@bibunitname{bu\the\@bibunitauxcnt}</code>

4.2 New cite commands

The main idea is to save old meaning of the command and then redefine it. At the beginning of a bibunit, the new commands are activated, at the end of a bibunit, the old commands are restored.

<code>\if@localcite</code>	Define a new if to switch between local and global cites, i.e., cites of references that occur in the local or global bibliography, resp.
	<code>5 \newif\if@localcite</code>
<code>\@localcitedefault</code>	Define and activate a default state, which is used to select whether references are generated for the local or <i>also</i> the global bibliography.
	<code>6 \def\@localcitedefault{\@localcitettrue}%</code>
	<code>7 \@localcitedefault</code>

```

\std@cite
\bu@cite 8 \let\std@cite\cite
          9 \def\bu@cite{%
          10 \@ifstar
          11   {\@localcitefalse\std@cite}%
          12   {\@localcitettrue\std@cite}}

\std@@citex
\bu@@citex 13 \let\std@@citex\@citex
            14 \def\bu@@citex[#1]#2{%
            15   \if@filesw
            16     \immediate\write\@bibunitaux{\string\citation{#2}}%
            17   \fi
            18   \if@localcite
            19     \if@filesw
            20       \@fileswfalse\std@@citex[#1]{#2}\@fileswtrue
            21     \else
            22       \std@@citex[#1]{#2}%
            23     \fi
            24   \else
            25     \std@@citex[#1]{#2}%
            26   \fi
            27   \@localcitedefault}

\std@nocite
\bu@nocite 28 \let\std@nocite\nocite
            29 \def\bu@nocite{%
            30   \@ifstar
            31   {\@localcitefalse\bu@nocite}%
            32   {\@localcitettrue\bu@nocite}}
            33 \def\bu@nocite#1{%
            34   \@bsphack
            35   \if@filesw
            36     \immediate\write\@bibunitaux{\string\citation{#1}}%
            37   \fi
            38   \@esphack
            39   \if@localcite
            40   \else
            41     \std@nocite{#1}%
            42   \fi
            43   \@localcitedefault}

```

4.3 New bibliography and bibliographystyle commands

In this paragraph no general description is given, rather an explanation why in the following two macros `\bu@bibliography` and `\bu@bibliographystyle` the command `\gdef\bu@bibdata` and `\gdef\bu@bibstyle` are written to the auxiliary file and are directly executed, too. The execution via the auxiliary file has been left for compatibility to older versions of bibunits: if you also have a global

bibliography, you can specify BIB_TE_X styles and files for the global bibliography (using `\bibliography` and `\bibliographystyle`, usually at the end of the document). In this case, writing the execution of `\gdef` to the auxiliary file ensures that in subsequent runs of L^AT_EX, defaults for the local bibliographies are generated as well. Without this, the user would have to explicitly specify BIB_TE_X files and styles for the local bibliographies using the starred form `\bibliography*` and `\bibliographystyle*`. On the other hand, with the direct execution of the `\gdef`'s, the starred form specify defaults which can be used already in the first run of L^AT_EX if put before the first bibunit.

```

\bu@bibdata Define data files store.
44 \def\bu@bibdata{}

\std@bibliography
\bu@bibliography 45 \let\std@bibliography\bibliography
46 \def\bu@bibliography{%
47   \ifstar
48     {\@localcitetrue\bu@bibliography}%
49     {\@localcitefalse\bu@bibliography}}

\@bu@bibliography
50 \def\@bu@bibliography#1{%
51   \if@filesw
52     \immediate\write\@auxout{\string\gdef\string\bu@bibdata{#1}}%
53   \fi
54   \gdef\bu@bibdata{#1}%
55   \if@localcite
56   \else
57     \std@bibliography{#1}%
58   \fi
59   \@localcitedefault}

\bu@bibstyle Define style store.
60 \def\bu@bibstyle{}

\std@bibliographystyle
\bu@bibliographystyle 61 \let\std@bibliographystyle\bibliographystyle
62 \def\bu@bibliographystyle{%
63   \ifstar
64     {\@localcitetrue\bu@bibliographystyle}%
65     {\@localcitefalse\bu@bibliographystyle}}

\@bu@bibliographystyle
66 \def\@bu@bibliographystyle#1{%
67   \if@filesw
68     \immediate\write\@auxout{\string\gdef\string\bu@bibstyle{#1}}%

```

```

69 \fi
70 \gdef\bu@bibstyle{#1}%
71 \if@localcite
72 \else
73 \std@bibliographystyle{#1}%
74 \fi
75 \@localcitedefault}

```

4.4 Bibunits

4.4.1 Bibunits by an extra environment

The starting of a bibunit is a two-step process: in the first step (macro `\bibunit`), all cite commands are redefined to trigger the second step, namely the actual starting of a bibunit (macro `\@startbibunit`). This may seem weird on the first sight, but has the advantage that no auxiliary files are created if a bibunit contains no `\cite`, `\cite*`, `\nocite` or `\nocite*` commands. To handle citations of the same reference in different bibliographies, the local auxiliary file is input at the beginning of each bibunit (in macro `\@startbibunit`), and the replacement text is also written to the *local* auxiliary file (in macro `\@putbib`).

```

\bibunit At the beginning of a bibunit, all cite command are set to their initializing ver-
        sions, and \@finishbibunit can relax, because currently no citation has appeared.
        Hence no local auxiliary file has been opened for the current bibunit, which has
        to be closed at the end of the bibunit.
        76 \def\bibunit{%
        77 \global\let\cite\@initcite
        78 \global\let\nocite\@initnocite
        79 \global\let\@finishbibunit\relax
        80 \ifnextchar[{\@bibunitx}{\@bibunitx[\bu@bibstyle]}}

\@bibunitx If the current bibunit has an extra bibstyle, it is saved in \@localbibstyle. (In
        version 1.0 a \bibstyle command has been directly written to the local auxiliary
        file. This is now done in \@startbibunit.)
        81 \def\@bibunitx[#1]{\gdef\@localbibstyle{#1}}

\endbibunit At the end of the bibunit, all cite commands are restored to their former meaning.
        Also the local auxiliary file is closed if one was opened.
        82 \def\endbibunit{%
        83 \global\let\cite\std@cite
        84 \global\let\@citex\std@@citex
        85 \global\let\nocite\std@nocite
        86 \@finishbibunit}

\@initcite Define initialization versions of the cite commands.
\@initnocite
        87 \def\@initcite{\@startbibunit\cite}
        88 \def\@initnocite{\@startbibunit\nocite}

```

`\@startbibunit` This command does all the necessary initialization for a bibunit: New versions of the cite commands are activated, the macro `\@finishbibunit` which is called by `\endbibunit` is let to close the local auxiliary file, the number of the bibunits is incremented. Most important, the local auxiliary file is read, so that citations of the same reference in different bibliographies are handled correctly for numerical citation schemes. Finally, the bibstyle is written to the local auxiliary file.

```

89 \def\@startbibunit{%
90   \global\let\cite\bu@cite
91   \global\let\@citex\bu@citex
92   \global\let\nocite\bu@nocite
93   \global\let\@finishbibunit\@finishstartedbibunit
94   \global\advance\@bibunitauxcnt 1
95   \if@filesw

```

Input the bibunit-auxfile and prevent the creation of unwanted spaces by setting `\endlinechar-1` (`\catcode'\^^M=9` would also work here). Thanks to Werner Jürgens for this hint.

```

96     {\endlinechar-1\@input{\@bibunitname.aux}}%
97     \immediate\openout\@bibunitaux\@bibunitname.aux
98     \immediate\write\@bibunitaux{\string\bibstyle{\@localbibstyle}}%
99   \fi}

```

When bibunits are specified by chapters or sections, a new chapter or section marks both the end of the old and the start of the new bibunit by calling `\endbibunit`. For the first chapter or section, the command `\@finishbibunit` has to be defined (to do nothing), which is done here.

```
100 \let\@finishbibunit\relax
```

`\@finishstartedbibunit` Close the local auxiliary file.

```

101 \def\@finishstartedbibunit{%
102   \if@filesw
103     \immediate\closeout\@bibunitaux
104   \fi}

```

4.4.2 Bibunits by chapters or sections

`\chapter` or `\section` substituting definition.

```

\old@bibunit
  \@bibunit 105 \let\old@bibunit\@gobble
\@endbibunit 106 \def\@bibunit{\endbibunit\bibunit\old@bibunit}
107 \def\@endbibunit{}

```

`\bibliographyunit`

```

108 \def\bibliographyunit{%
109   \@endbibunit
110   \@ifnextchar[{\@bibliographyunit}{%
111     \global\let\old@bibunit\@gobble

```

```

112 \global\let\bibliography\std@bibliography
113 \global\let\bibliographystyle\std@bibliographystyle
114 \endbibunit
115 \gdef\@endbibunit{}}

```

`\@bibliographyunit`

```

116 \def\@bibliographyunit[#1]{%
117 \global\let\bibliography\bu@bibliography
118 \global\let\bibliographystyle\bu@bibliographystyle
119 \global\let\old@bibunit#1
120 \global\let#1\@bibunit
121 \gdef\@endbibunit{\global\let#1\old@bibunit}}

```

4.5 Insert local bibliography: putbib

`\putbib`

```

122 \def\putbib{\@ifnextchar [{\@putbib}{\@putbib[\bu@bibdata]}}

```

`\@putbib`

The package allows the same cited item to occur in more than one bibliography. Therefore the text which actually replaces the `\cite` has to be kept locally, too. This text is created by the `\bibitem` commands in the `.bbl` file. The `\bibitem` commands invoke macros (`\@lbibitem` or `\@bibitem`) which by default write to `\@auxout`. We locally set `\@auxout` to `\@bibunitaux` so that they now write in the local auxiliary file of the bibunit.

```

123 \def\@putbib[#1]{%
124 \if@filesw
125 \immediate\write\@bibunitaux{\string\bibdata{#1}}%
126 \fi
127 {\let\@auxout\@bibunitaux \@input{\@bibunitname.bbl}}

```

4.6 Allow commands to be used not only in preamble

To handle citations of the same cited item in more than one bibliography, replacement text for the citation is written to the local auxiliary file. In subsequent runs of `LATEX`, this file is read at the beginning of the bibunit.

`\remequivalent`

`\plugh`

`\hgulp`

The local auxiliary file contains `\babcite` commands, which can be used only in the preamble in some older version of `LATEX`. For compatibility with these versions, macros to remove an item from a list are provided (see the `TEXbook`, p. 380; the only change is the separator `\do` instead of `\`).

```

128 \def\remequivalent#1\from#2{%
129 \let\given=#1%
130 \ifx#2\empty
131 \else
132 \edef#2{\expandafter\plugh#2\plugh}%
133 \fi}
134 \def\plugh\do#1#2{%
135 \ifx#1\given

```

```

136 \else
137   \noexpand\do\noexpand#1%
138 \fi
139 \ifx#2\plugh
140   \hgulp\fi\plugh#2}
141 \def\hgulp\fi\plugh\plugh{\fi}

```

Now we are ready to remove `\bibcite` from the `\@preamblecmds` list.

```

142 \remequivalent\bibcite\from\@preamblecmds

```

The command `\bibcite` calls `\@newl@bel` to create new labels. This is by default only allowed as a preamble command and would cause a ‘Can be used only in preamble’ warning. Instead of removing `\@newl@bel` from the list of preamble command (like `\bibcite`), we redefine `\bibcite` to call a modified version of `\@newl@bel` which issues no warning. We decided not to give even an info, because multiple cited references should be perfectly normal to bibunits.

```

\bibcite
\bu@newl@bel 143 \def\bibcite{\bu@newl@bel b}
144 \def\bu@newl@bel#1#2#3{\global\@namedef{#1@#2}{#3}}

145 </package>

```