

The `bibunits` Package

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

v2.4 2004/05/12

Abstract

The `bibunits` package allows separate bibliographies for different units or parts of the text. The units can be chapters, sections or `bibunit` environments. The package separates the citations of each unit of text into a local auxiliary 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 package is compatible with a wide variety of packages, including, but not limited to, `cite`, `jurabib`, `natbib`, `overcite` and `KOMA-SCRIPT` classes. It is based on the `bibunits` style by José Alberto Fernández.

1 Introduction

Suppose, you have different units or parts within your document, and each unit should have its own bibliography. In this case, you can use the `bibunits` package to do the work.

Let's first consider a simple example to see how `bibunits` works (see Fig. 1). We specify two `bibunits` using the `bibunit` environment. The optional parameter specifies the bibliography style to be used within the `bibunits` bibliography, in this case `plain` for the first and `alpha` for the second `bibunit`. Citations can be made as usual. Similar to the `\bibliography` command, you have to specify where the bibliography should appear within a `bibunit`. This is done using `\putbib`. The optional parameter specifies the `BIBTEX` files to be used for generating the `bibunits` bibliography, in this case `lit`.

For each `bibunit`, there is now a separate auxiliary file `bu<i>i</i>.aux` to be processed by `BIBTEX`. In this example, we have two `bibunits` with auxiliary files `bu1.aux` and `bu2.aux`. To process your document, three runs of `LATEX` and two runs of `BIBTEX` are required.

```

\documentclass{article}
\usepackage{bibunits}

\begin{document}
\begin{bibunit}[plain]
References to the \TeX book \cite{Knuth:1991}
and to Lamport's \LaTeX\ book, which appears
only in the references\nocite{Lamport:1994}.
\putbib[lit]
\end{bibunit}

\begin{bibunit}[alpha]
A new bibunit with a different style and a cite
of a Postscript tutorial \cite{Adobe:1985}.
\putbib[lit]
\end{bibunit}

\end{document}

```

References to the T_EXbook [1] and to Lamport's L^AT_EX book, which appears only in the references.

References

- [1] Donald E. Knuth. *The T_EX book*. Addison-Wesley, Reading, Massachusetts, 1991.
- [2] Leslie Lamport. *L^AT_EX: A Document Preparation System*. Addison-Wesley, Reading, Massachusetts, 2 edition, 1994.

A new bibunit with a different style and a cite of a Postscript tutorial [Ado85].

References

- [Ado85] Adobe System Incorporated. *Postscript Language Tutorial and Cookbook*. Addison-Wesley, Reading, Massachusetts, 1985.
-

Figure 1: Example input and output.

```

latex mydoc
bibtex bu1
bibtex bu2
latex mydoc
latex mydoc

```

This example should give a pretty good impression of what principally can be done with `bibunits` and how it can be done. But `bibunits` allows for many more features, e.g., `bibunits` can be specified by `\sections`, a global bibliography can be used, and `cite` command can be placed in both the local and the global bibliography at the same time, among other features. All these features and associated commands are described in the next section.

2 Usage Notes

There are two ways to specify the units for which a bibliography is generated. You can either define your own `bibunit` environment, or you can specify `bibunits` for the L^AT_EX units `\chapter` or `\section`. Both ways are mutually exclusive, meaning that you can choose only one and cannot mix them.

`bibunit` Use the environment `\begin{bibunit}[\langle style \rangle]` to create a `bibunit`. The optional argument `\langle style \rangle` specifies a bibliography style. If the optional argument `\langle style \rangle` is omitted, the default style is used. How to define a default style is explained below within this section.

`\bibliographyunit` Alternatively, use `\bibliographyunit[⟨unit⟩]`, where `⟨unit⟩` can be `\chapter` or `\section` to specify for which document unit a bibliography is generated, namely for every chapter or for every section. Use `\bibliographyunit` with no arguments to deactivate bibliography units. By default `\bibliographyunit` is deactivated. The command `\bibliographyunit` has to be issued after `\begin{document}`.

`\cite*` You can create a global bibliography as usual with the commands
`\nocite*` `\bibliography[⟨BibTeX files⟩]` and `\bibliographystyle[⟨style⟩]`. Use `\cite` and `\nocite` to generate citations that appear in the local bibliography. Use `\cite*` and `\nocite*` inside a unit to generate citations for both the local and global bibliography.

`\putbib` Use `\putbib[⟨BibTeX files⟩]` to specify where the bibliography should appear within a bibunit. If the optional argument is omitted, `\putbib` uses the default BibTeX files. How to define a default files is explained below within this section.

`\defaultbibliography` Use `\defaultbibliography{⟨BibTeX files⟩}` and
`\defaultbibliographystyle` `\defaultbibliographystyle{⟨style⟩}` to specify default BibTeX data files and style to be used in the local bibliographies. If no defaults are specified, `\jobname` is used as the default bibliography file and `plain` is used as the default bibliography style.

For compatibility to pre-v.2.3 versions of `bibunits`, two other ways to define default BibTeX styles and files for the local units are possible, but their use is not recommended.

If `\bibliographyunit` is active, there are two other ways to define default BibTeX styles and files for the local units: either by defining a global bibliography using `\bibliography` and `\bibliographystyle` or by using starred versions `\bibliography*` and `\bibliographystyle*`, as detailed below.

`\bibliography` If `\bibliographyunit` is active, the commands `\bibliography[⟨BibTeX files⟩]` and
`\bibliographystyle` `\bibliographystyle[⟨style⟩]` also specify the BibTeX files and style to be used by default in the local units. The commands `\bibliography` and `\bibliographystyle` have to be issued after `\begin{document}`.

`\bibliography*` If `\bibliographyunit` is active, you can use starred forms `\bibliography*{⟨BibTeX files⟩}`
`\bibliographystyle*` and `\bibliographystyle*{⟨style⟩}` to specify the defaults for the local units only. These commands do not generate a global bibliography or any information for the global bibliography. The commands `\bibliography*` and `\bibliographystyle*` have to be issued after `\begin{document}`.

Note that the old pre-v.2.3 ways to define default style and BibTeX files can only be specified while `\bibliographyunit` is active. Thus, if you do want default styles but `bibunit` environments instead of `\bibliographyunits`, you have to issue a dummy `\bibliographyunit[⟨section⟩]`, then specify defaults `\bibliography*{⟨BibTeX files⟩}` and `\bibliographystyle*{⟨style⟩}` and finally switch off the generation of `bibunits` by sections using `\bibliographyunit`.

2.1 Package options

`globalcitecopy` Create an entry in the global bibliography for each cite command within a bibunit. Default is to create an entry only for the local bibliography.

`labelstoglobalaux` Write the replacement text for a label inside a bibunit to the global auxiliary file. Default is to write the labels to the local auxiliary file. The option `labelstoglobalaux` is useful in some situations, e.g.,

- to resolve the references when a bibunit bibliography is `\included`.

- to resolve citations within a moving argument like `\section`, which otherwise would result in an ‘Undefined reference’ error.

Note that this option results in wrong labels for numerical reference schemes if the same cited item appears at different positions in more than one bibliography (cf. Sec. 2.3).

`sectionbib` Let bibliography titles appear at the section level. Useful for chapters as bibliography units.

`subsectionbib` Let bibliography titles appear at the subsection level. Useful for sections as bibliography units.

2.2 BIBTEX processing

For each bibunit, in sequence, there is now a corresponding file `bu⟨i⟩.aux` that needs to be compiled through BIBTEX. Suppose your document has $\langle n \rangle$ different bibunits, you must now invoke BIBTEX on `bu1, …, bu⟨n⟩`. 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 you don't have the `basename` command, you can use an alternative script where the suffix `.aux` is removed using `sed`.

```
#!/bin/csh
foreach auxfile (bu*.aux)
  set auxfilebasename = `echo $auxfile | sed s/.aux//`
  echo bibtex $auxfilebasename
  bibtex $auxfilebasename
end
```

If by some strange coincidence you have named some of your files `bu⟨i⟩.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.

2.3 Limitations

- A citation within a moving argument in a bibunit cannot be resolved outside this bibunit, because the label is local to the bibunit. This occurs, e.g., for a `\cite` within `\section`, which is undefined in the table of contents. You can use the option `labelstoglobalaux` to resolve the citations, but this has two side effects.
 - Since the `cite` command also appears globally in the table of contents, it generates a `\citation` command for the *global* auxiliary file. As a result, another `BIBTEX` run causes this previously local reference to creep into the global bibliography.
 - Since the replacement text for all cited items is now global, the same cited item has always the same replacement text, regardless whether it appears at different positions within different bibunits. This results in wrong labels for numerical reference schemes if the same cited item appears at different positions within the bibliographies.
- When using `natbib`, `\cite*` has the meaning as defined by `natbib`, namely to force full citation of multiple authors. To get the behavior as defined by `bibunits` (without `natbib`), use an accompanying `\nocite` outside of a bibunit. Alternatively, if you want *every* local item to appear also in the global bibliography, you can use `bibunits` with option `globalcitecopy`.
- With `natbib`, you cannot use numerical and author-year reference schemes together. Typographically, this is regarded bad style anyway.

3 Examples

3.1 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{Lampport:1994} more text more citations
  \putbib[texlit]
\end{bibunit}
some text between the units
\begin{bibunit}[abbrv]
  some text \cite{Knuth:1991} 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 using `\defaultbibliography` and `\defaultbibliographystyle`.

```

\documentclass{article}
\usepackage{bibunits}

\begin{document}
\defaultbibliography{texlit}
\defaultbibliographystyle{plain}

\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, e.g., the class `article`, the heading of the bibliography has the same font 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. This can be done by passing the option `subsectionbib` to `bibunits`.

```

\documentclass{article}
\usepackage[subsectionbib]{bibunits}

```

Similarly, if you use a class such as `book` where the bibliography headings appear at the chapter level by default, headings of the local bibliographies can be changed to appear at the section level by passing the option `sectionbib` to `bibunits`.

```

\documentclass{book}
\usepackage[sectionbib]{bibunits}

```

3.2 Bibunits by chapters or sections

You can also define bibliographies for every chapter or section. In this case, the redefinition of `\thebibliography` by passing options `sectionbib` resp. `subsectionbib` 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 opened 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 BIB_TE_X files and styles. Note that you have to specify the `\defaultbibliographystyle` *before* the corresponding section.

```
\documentclass{article}
\usepackage[subsubsectionbib]{bibunits}

\begin{document}
\bibliographyunit[\section]
\defaultbibliographystyle{plain}

\section{First section}
  some text \cite{Lamport:1994} more text more citations
  \putbib[teXlit]
some text between the units
\defaultbibliographystyle{abbrv}
\section{Second and last section}
  some text \cite{Knuth:1991} more text more citations
  \putbib[lit]
\end{document}
```

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

```
\documentclass{article}
\usepackage[subsubsectionbib]{bibunits}

\begin{document}
\bibliographyunit[\section]
\defaultbibliography{teXlit}
\defaultbibliographystyle{plain}

\section{First section}
  some text \cite{Lamport: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}
```

3.3 Bibunits and a global bibliography

In all four examples, one can specify a global bibliography and its style with the usual L^AT_EX 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 `\bibliographyunit` is active, *and* you use a global bibliography, *and* you use the same BIB_TE_X file and style for all (local as well as global) bibliographies,

then the `\defaultbibliography` and `\defaultbibliographystyle` 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.

The font size used to typeset the global bibliography is the same as for the local bibliographies. To change this, switch off the bibliography units by chapters or sections using `\bibliographyunit` and change `\thebibliography` to its original meaning just before the global bibliography.

```
\bibliographyunit
\let\thebibliography\stdthebibliography
\bibliographystyle{plain} % global bibliography
\bibliography{texlit}    % global bibliography
```

4 Acknowledgments

The author thanks José Alberto Fernández for the coding of `bibunits` v1.0. Further, contributions of code fragments from various people are gratefully acknowledged (in alphabetical order): Battista Benciolini for the patch to allow `putbib` to appear before the first occurrence of a cite within a `bibunit`; Jens Berger for the `jurabib` fragment of the `babel` compatibility code; Jason Harrison for an alternative script to call \BIBTeX for each local auxiliary file; Werner Jürgens for the hint on how to suppress unwanted spaces; Frank Mittelbach for a code fragment how to make `bibunits` “Label(s) may have changed.” warning appear at the end of a \LaTeX run; Andrew E. Schulman for an early version of the `natbib` compatibility patch; Stefan Ullrich for an early version of the `jurabib` compatibility patch (communicated by Jens Berger) and the `mparhack` compatibility patch. Finally, the author acknowledges the contributions of numerous people, in particular Frank Mittelbach, whose suggestions and bug reports have helped to improve `bibunits`.

5 The Macros

```
1 <*package>
```

5.1 Allocation

```
\@bu@tempcnta [Allocation of counter removed.]
```

5.2 Option handling

```
\iflabelstoglobalaux Define a new if to switch between global definition of labels within a bibliography
and local labels (default). Global labels can be activated with option labelstoglobalaux.
```

```
2 \newif\iflabelstoglobalaux \labelstoglobalauxfalse
3 \DeclareOption{labelstoglobalaux}{\labelstoglobalauxtrue}
```

`\ifglobalcitecopy` Define a new if to switch between entering of cite commands within a bibunit also to the global bibliography or only to the local bibliography (default). Entering of cite commands also to the global bibliography can be activated with option `globalcitecopy`.

```
4 \newif\ifglobalcitecopy
5 \globalcitecopyfalse
6 \DeclareOption{globalcitecopy}{\globalcitecopytrue}
```

`sectionbib` Let bibliography titles appear at the section level. Useful for chapters as bibliography units.

```
7 \DeclareOption{sectionbib}{%
8   \AtBeginDocument{%
9     \let\stdthebibliography\thebibliography
10    \def\thebibliography{%
11      \let\chapter\section
12      \stdthebibliography}}
```

`subsectionbib` Let bibliography titles appear at the subsection level. Useful for sections as bibliography units.

```
13 \DeclareOption{subsectionbib}{%
14   \AtBeginDocument{%
15     \let\stdthebibliography\thebibliography
16     \def\thebibliography{%
17       \let\chapter\subsection
18       \let\section\subsection
19       \stdthebibliography}}
```

Finally, process all package options.

```
20 \ProcessOptions
```

5.3 An auxiliary file for each bibunit is provided

`\@bibunitaux` Define the file descriptor `\@bibunitaux` of the auxiliary file that is generated for each bibunit.

```
21 \newwrite\@bibunitaux
```

`\@bibunitauxcnt` Define the counter for the bibunits and initialize it with zero.

```
22 \newcount\@bibunitauxcnt \@bibunitauxcnt=0
```

`\@bibunitname` The command `\@bibunitname` sets the basename of the auxiliary files that are created for each bibunit to `bu` (bu for bibunits), followed by the current number of the bibunit, `\the\@bibunitauxcnt`. (In version 1.0 of bibunits there was no such command, and the command sequence `\jobname.\the\@bibunitauxcnt` was directly used.)

```
23 \def\@bibunitname{bu\the\@bibunitauxcnt}
```

5.4 New cite commands

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

`\if@localcite` Define a new if to store if a macro is evoked with or without a star. This is used by cite commands to switch between local and global cites, i.e., cites of references that occur in the local or also in the global bibliography. The is also a starred version of `\bibliopography` and `\bibliopographystyle` to enter default BibTeX data files and styles for the bibunits only.

```
24 \newif\if@starredversion
```

`\@localcitedefault` [Macro deleted.]

```
\std@cite
\bu@cite 25 \let\std@cite\cite
26 \DeclareRobustCommand\bu@cite{%
27   \ifstar
28     {\@starredversiontrue\std@cite}%
29     {\@starredversionfalse\std@cite}%
30 }
```

`\std@@citex` Define the bibunits macro `\bu@@citex` replacing the internal macro `\@citex`.
`\bu@@citex` Macro `\@citex` is evoked by the `\cite` and its various derivatives defined by other packages. Basically, `\bu@@citex` the following actions take place. The macro `\startbibunitorrelax` ensures that auxiliary files are created only for non-empty bibunits. The macro `\leavevmode` are added to fix a problem that occurs in some special situations when a `\paragraph` within a bibunit is immediately followed by a cite command, as explained below. Next, the standard version of `\@citex`, namely `\std@@citex` is evoked with `\@auxout` locally changed, so that citations are written to the local auxiliary file of each bibunit. Finally, if option `globalcitecopy` is active or the starred form of the cite command is evoked, i.e., `\if@starredversion` is true, citations are *also* written to the global auxiliary file, so that citations appear in both the local as well as the global bibliography at the same time.

After this general description, the use of `\leavevmode` is further detailed. The `\leavevmode` commands are added to fix a problem that occurs when a `\paragraph` within a bibunit is immediately followed by a cite command, and within `\paragraph` something is written to `\@auxout`, e.g., an entry for the table of contents or the index. In this cases, the entry go to the local auxiliary file of the corresponding bibunit instead of the standard auxiliary file `\@auxout`. The reason is that the argument of `\paragraph` is stored in an `\everypar`, which is evaluated later. In case of a cite command that immediately follows `\paragraph`, the evaluation occurs within the scope of the cite, where the `\@auxout` is let to the local auxiliary file of the corresponding bibunit. Consequently, the entry is

written to the wrong, i.e., local, auxiliary file. We add an `\leavevmode` to evaluate `\everypar` before `\@auxout` is redefined. Alternatively, the typesetting of `\paragraph` in `\@xsect` of the L^AT_EX kernel might be revised.

Additional to the new standard definition of `\@citex`, different versions have to be defined depending other packages loaded, namely `natbib`, `overcite` and `jurabib`. These packages are mutually exclusive, with the priority as defined by their order in the previous sentence. That means, e.g., if you have chosen `natbib` together with `bibunits`, you cannot use `overcite` or `jurabib`.

All definitions are done `\AtBeginDocument` so that packages loaded after `bibunits` can be discovered.

```
31 \AtBeginDocument{%
```

Several modifications and additions are necessary if package `natbib` is loaded: First macro `\NAT@set@cites` is used to invoke `natbib`'s cite commands. This is only necessary if these commands have not been set, i.e. in case `bibunits` is evoked before `natbib`, but does not do any harm otherwise. Second macro `\bu@@citex` is changed as described below, and the various versions of `natbib`'s cite commands are save in a `\std...` version.

The definition of `\bu@@citex` is changed in the following ways: i) the `\let` of `\@auxout` is encapsulated using `\begingroup` and `\endgroup` instead of braces, since braces results in an error; ii) macro `\std@@citex` has two optional arguments instead of one; iii) the code of `\if@starredversion` is skipped in favor of the meaning of a starred cite in `natbib`, namely to force full listing of multiple authors.

```
32 \@ifpackageloaded{natbib}%
33   {%
34     \NAT@set@cites \let\std@@citex\@citex
35     \def\bu@@citex[#1][#2]#3{%
36       \@startbibunitorelax
37       \leavevmode
38       \begingroup\let\@auxout\@bibunitaux\std@@citex[#1][#2]{#3}\endgroup
39       \ifglobalciterecopy
40         \std@nocite{#3}%
41       \fi
42     }%
43   }%
```

If `natbib` is not loaded, we check for `cite`. In this case, an additional command `\@citew` has to be redefined which handles `\cite` without an argument. Further, an internal command of `cite`, namely `\@nocite` is used to write to the standard auxiliary file.

```
44   {%
45     \@ifpackageloaded{cite}%
46     {%
47       \let\std@@citew\@citew
48       \def\bu@@citew#1{%
49         \@startbibunitorelax
50         \leavevmode
```

```

51     {\let\@auxout\@bibunitaux \std@@citew{#1}}%
52     \ifglobalcitecopy
53       \@nocite{#1}%
54     \else
55       \if@starredversion
56         \@nocite{#1}%
57       \fi
58     \fi
59   }
60   \let\std@@citex\@citex
61   \def\bu@@citex[#1]#2{%
62     \startbibunitorrelax
63     \leavevmode
64     {\let\@auxout\@bibunitaux \std@@citex[#1]{#2}}%
65     \ifglobalcitecopy
66       \@nocite{#2}%
67     \else
68       \if@starredversion
69         \@nocite{#2}%
70       \fi
71     \fi
72   }
73 }%

```

Next we check if `overcite` is loaded. This is done for compatibility reasons only, since `overcite` is integrated into `cite` since version 4.0.—We give a warning that an obsolete package has been loaded and do the same definitions as for `cite`.

```

74   {%
75     \ifpackageloaded{overcite}%
76     {%
77       \PackageWarningNoLine{bibunits}%
78       {Obsolete package overcite loaded. Use package cite which
79         option [superscript] instead}%
80     \let\std@@citew\@citew
81     \def\bu@@citew#1{%
82       \startbibunitorrelax
83       \leavevmode
84       {\let\@auxout\@bibunitaux \std@@citew{#1}}%
85       \ifglobalcitecopy
86         \@nocite{#1}%
87       \else
88         \if@starredversion
89           \@nocite{#1}%
90         \fi
91       \fi
92     }
93     \let\std@@citex\@citex
94     \def\bu@@citex[#1]#2{%
95       \startbibunitorrelax
96       \leavevmode

```

```

97         {\let\@auxout\@bibunitaux \std@citex[#1]{#2}}%
98         \ifglobalciterecopy
99           \nocite{#2}%
100        \else
101          \ifstarredversion
102            \nocite{#2}%
103          \fi
104        \fi
105      }
106    }%

```

If neither `natbib` nor `cite` nor `overcite` are loaded, we check for `jurabib`. In this case, the definition of `\std@citex` differs from the default version by a second optional argument, similar to `natbib`.

```

107    {%
108      \ifpackageloaded{jurabib}%
109      {%
110        \let\std@citex\@citex
111        \def\bu@citex[#1][#2]#3{%
112          \@startbibunitorrelax
113          \leavevmode
114          {\let\@auxout\@bibunitaux \std@citex[#1][#2]{#3}}%
115          \ifglobalciterecopy
116            \std@nocite{#3}%
117          \else
118            \ifstarredversion
119              \std@nocite{#3}%
120            \fi
121          \fi
122        }%
123      }%

```

Finally, the default definition.

```

124    {%
125      \let\std@citex\@citex
126      \def\bu@citex[#1]#2{%
127        \@startbibunitorrelax
128        \leavevmode
129        {\let\@auxout\@bibunitaux \std@citex[#1]{#2}}%
130        \ifglobalciterecopy
131          \std@nocite{#2}%
132        \else
133          \ifstarredversion
134            \std@nocite{#2}%
135          \fi
136        \fi
137      }%
138    }%
139  }%
140 }%

```

```

141 }%
142 }%

\std@nocite
\bu@nocite 143 \let\std@nocite\nocite
144 \def\bu@nocite{%
145   \@ifstar
146     {\@starredversiontrue\@bu@nocite}%
147     {\@starredversionfalse\@bu@nocite}%
148 }

\@bu@nocite
149 \def\@bu@nocite#1{%
150   \@startbibunitrelax
151   {\let\@auxout\@bibunitaux \std@nocite{#1}}%
152   \ifglobalcitecopy
153     \std@nocite{#1}%
154   \else
155     \if@starredversion
156       \std@nocite{#1}%
157     \fi
158   \fi
159 }

```

5.5 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 through the auxiliary file has been left for compatibility to older versions of `bibunits`: if you also have a global bibliography, you can specify `BIBTEX` 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 `LATEX` defaults for the local bibliographies are generated as well. Without this, the user would have to explicitly specify `BIBTEX` files and styles for the local bibliographies using the starred forms `\bibliography*` and `\bibliographystyle*`. On the other hand, with the direct execution of the `\gdef` commands, the starred forms specify defaults which can be used already in the first run of `LATEX` if put before the first bibunit.

```

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

\orig@bibliography If \labelstoglobalaux is false, define a new version of \bibliography where the
\ Bibliography replacement text for the labels is written to a new file bu.aux. This file is input
\std@bibliography at the end of each bibunit to restore the global labels which are overwritten if
a global item is also cited within a bibunit. Input of the main auxiliary file is

```

not appropriate, because this file contains additional material which should not be input more than once, e.g., commands which generate entries for the table of contents.

The definition of `\std@bibliography` is used to define a version of `\bibliography` with enhanced functionality `\bu@bibliography`.

```

161 \AtBeginDocument{%
162   \iflabelstoglobalaux
163   \else
164     \let\orig@bibliography\bibliography
165     \def\bibliography#1{%
166       \if@filesw
167         \immediate\openout\@bibunitaux bu.aux
168         \immediate\write\@mainaux{\string\@input{bu.aux}}%
169       \fi
170       \orig@bibliography{#1}%
171     \if@filesw
172       \immediate\closeout\@bibunitaux
173     \fi
174   }%
175 \fi
176 \let\std@bibliography\bibliography
177 }

```

`\bu@bibliography`

```

178 \def\bu@bibliography{%
179   \@ifstar
180   {\@starredversiontrue\@bu@bibliography}%
181   {\@starredversionfalse\@bu@bibliography}%
182 }

```

`\@bu@bibliography`

```

183 \def\@bu@bibliography#1{%
184   \defaultbibliography{#1}%
185   \if@starredversion
186   \else
187     \std@bibliography{#1}%
188   \fi
189 }

```

`\defaultbibliography` `\defaultbibliography` allows to define default bibliography files whether `bibunits` is active or not. Does the same as `\bibliography*` when `bibunits` is active.

```

190 \def\defaultbibliography#1{%
191   \if@filesw
192     \immediate\write\@auxout{\string\gdef\string\bu@bibdata{#1}}%
193   \fi
194   \gdef\bu@bibdata{#1}%
195 }

```

```

\bu@bibstyle Define style store.
196 \def\bu@bibstyle{plain}

\std@bibliographystyle
\bu@bibliographystyle 197 \let\std@bibliographystyle\bibliographystyle
198 \def\bu@bibliographystyle{%
199   \@ifstar
200   {\@starredversiontrue\@bu@bibliographystyle}%
201   {\@starredversionfalse\@bu@bibliographystyle}%
202 }

\@bu@bibliographystyle
203 \def\@bu@bibliographystyle#1{%
204   \defaultbibliographystyle{#1}%
205   \if@starredversion
206   \else
207   \std@bibliographystyle{#1}%
208   \fi
209 }

\defaultbibliographystyle \defaultbibliographystyle allows to define a default bibliography style whether
bibunits is active or not. Does the same as \bibliographystyle* when bibunits
is active.

210 \def\defaultbibliographystyle#1{%
211   \if@filesw
212   \immediate\write\@auxout{\string\gdef\string\bu@bibstyle{#1}}%
213   \fi
214   \gdef\bu@bibstyle{#1}%
215 }

```

5.6 Bibunits

5.6.1 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 versions, 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.

```
216 \def\bibunit{%
```

```

217 \global\let\cite\bu@cite
218 \global\let\@citex\bu@@citex
219 \global\let\@citew\bu@@citew
220 \global\let\nocite\bu@nocite
221 \global\let\@startbibunit\relax\@startbibunit
222 \global\let\@finishbibunit\relax
223 \@ifnextchar[{\@bibunitx}{\@bibunitx[\bu@bibstyle]}%
224 }

```

`\@bibunitx` If the current bibunit has an extra bibstyle, it is saved in `\@localbibstyle`. (In version 1.0a `\bibstyle` command has been directly written to the local auxiliary file. This is now done in `\@startbibunit`.)

```

225 \def\@bibunitx[#1]{\gdef\@localbibstyle{#1}}

```

`\endbibunit` At the end of the bibunit, all cite commands are restored to their former meaning (`\@citew` is used by package `overcite`). Also the local auxiliary file is closed if it has been opened. Finally, the replacement text for the global citation labels (if exists) are restored using `\@input`.

```

226 \def\endbibunit{%
227 \global\let\cite\std@cite
228 \global\let\@citex\std@@citex
229 \global\let\@citew\std@@citew
230 \global\let\nocite\std@nocite
231 \@finishbibunit
232 \@input{bu.aux}%
233 }

```

`\@startbibunit` This command does all the necessary initialization for a bibunit: New versions of the cite commands are activated (`\@citew` is used by package `overcite`), 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.

```

234 \def\@startbibunit{%
235 \global\let\@startbibunit\relax\relax
236 \global\let\@finishbibunit\@finishstartedbibunit
237 \global\advance\@bibunitauxcnt 1
238 \if@filesw

```

Input the local auxiliary file and prevent the creation of unwanted spaces by setting `\endlinechar-1` (`\catcode'\^M=9` would also work here).

```

239 \endlinechar-1
240 \@input{\@bibunitname.aux}%
241 \immediate\openout\@bibunitaux\@bibunitname.aux
242 \immediate\write\@bibunitaux{\string\bibstyle{\@localbibstyle}}%
243 \fi
244 }

```

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.

```
245 \let\@finishbibunit\relax
```

`\@finishstartedbibunit` Close the local auxiliary file.

```
246 \def\@finishstartedbibunit{%
247   \if@filesw
248     \immediate\closeout\@bibunitaux
249   \fi
```

Check for if the labels have changed and a “Rerun to get cross-references right” warning needs to be given. The mechanism is the same as in `\enddocument`: redefine `\bibcite` to check if the labels have changed and then read the auxiliary file. Since we have changed `\bibcite` to call directly `\@namedef`, we use a variant `\@xtestdef` of L^AT_EX’s `\@testdef` for this job. Also, we do not have to `\makeatletter` before reading `\@bibunitaux`, because this file is supposed to contain only bibliography related macros without any `@`’s. But we do have to change `\endlinechar` as above to prevent the creation of unwanted spaces.

```
250   \ifx\bibunits@rerun@warning\@empty
251     \if@filesw
252       {\endlinechar-1
253        \let\@namedef\@xtestdef
254        \@input{\@bibunitname.aux}}%
255     \fi
256   \fi
257 }
```

`\bibunits@rerunwarning` Initialize `\bibunits@rerun@warning` as empty and call it `\AtEndDocument`. The definition of `\bibunits@rerun@warning` may be changed in macro `\@xtestdef`.

```
258 \let\bibunits@rerun@warning\@empty
259 \AtEndDocument\bibunits@rerun@warning
```

`\@xtestdef` Check if labels created by `\bibcite` in the local auxiliary files may have have changed. Analogous to `\@testdef` of L^AT_EX.

```
260 \def\@xtestdef #1#2{%
261   \def\reserved@a{#2}%
262   \expandafter\ifx\csname#1\endcsname\reserved@a
263   \else
264     \gdef\bibunits@rerun@warning{%
265       \PackageWarningNoLine{bibunits}{Label(s) may have changed.
266         Rerun to get cross-references right}}%
267   \fi
268 }
```

5.6.2 Bibunits by chapters or sections

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

```
\old@bibunit
  \@bibunit 269 \let\old@bibunit\@gobble
\@endbibunit 270 \def\@bibunit{\endbibunit\bibunit\old@bibunit}
                271 \def\@endbibunit{}

\bibliographyunit
272 \def\bibliographyunit{%
273   \@endbibunit
274   \@ifnextchar[{\@bibliographyunit}{%
275     \global\let\old@bibunit\@gobble
276     \global\let\bibliography\std@bibliography
277     \global\let\bibliographystyle\std@bibliographystyle
278     \endbibunit
279     \gdef\@endbibunit{}}}%
280 }

\@bibliographyunit
281 \def\@bibliographyunit[#1]{%
282   \global\let\bibliography\bu@bibliography
283   \global\let\bibliographystyle\bu@bibliographystyle
284   \global\let\old@bibunit#1
285   \global\let#1\@bibunit
286   \gdef\@endbibunit{\global\let#1\old@bibunit}%
287 }

\putbib If the BIBTEX data file is given as an optional argument, evoke \@putbib which
        uses this argument, otherwise call \@putbib with the default BIBTEX data file
        stored in \bu@bibdata.
288 \def\putbib{\@ifnextchar[{\@putbib}{\@putbib[\bu@bibdata]}}

\@putbib Similar to \bibliography, \@putbib writes the BIBTEX data and \@inputs the
        typeset bibliography \@bibunitname.bbl. Macro \@startbibunit or \relax triggers
        a new bibunit and allows \putbib to be issued before the first cite command
        within a bibunit.
289 \def\@putbib[#1]{%
290   \@startbibunit or relax
291   \if@filesw
292     \immediate\write\@bibunitaux{\string\bibdata{#1}}%
293   \fi
294   \@input@{\@bibunitname.bbl}%
295 }

\bu@addtocontents [Macro removed.]
```

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 can be switched off using `labelstoglobalaux`.

The text which replaces the `\cite` is created by the `\bibitem` commands in the `.bbl` file which is `\inputed` in `\@putbib`. The `\bibitem` macro invokes macros `\@lbibitem` or `\@bibitem` which write to `\@auxout`. By default, i.e., without option `labelstoglobalaux` these commands are replaced by `bibunits` version to write to the local auxiliary file of the `bibunit`. The former meaning of `\@auxout` is saved in `\temp@auxout` to be restored afterwards.

Note that some packages, like `mparhack` or KOMA-SCRIPT classes like `scartcl` with option `bibtoc` also write material to `\@auxout` while inside `\thebibliography` within the `.bbl` file. Therefore, redefining `\@auxout` in `\putbib` would result in erroneously writing this material to the local auxiliary file. Therefore, only the commands which actually write, i.e., `\@bibitem` and `\@lbibitem`, are redefined to write to the local auxiliary file.

Note that the redefined command `\bibliography` which generates a global bibliography uses the same mechanism to write the replacement text for the `cite` commands to `bu.aux`.

```

\std@bibitem Define new versions of \@bibitem and \@lbibitem which write to \bibunitaux
\std@lbibitem instead of \@auxout. Encapsulation with braces instead of the two lets results in
  \@bibitem \@noitemerr for the thebibliography environment.
  \@lbibitem
296 \AtBeginDocument{%
297   \iflabelstoglobalaux
298   \else
299     \let\std@bibitem\@bibitem
300     \let\std@lbibitem\@lbibitem
301     \def\@bibitem#1{%
302       \let\temp@auxout\@auxout
303       \let\@auxout\@bibunitaux
304       \std@bibitem{#1}%
305       \let\@auxout\temp@auxout
306     }
307     \def\@lbibitem[#1]#2{%
308       \let\temp@auxout\@auxout
309       \let\@auxout\@bibunitaux
310       \std@lbibitem[#1]{#2}%
311       \let\@auxout\temp@auxout
312     }
313   \fi
314 }

```

5.8 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 L^AT_EX, this file is read at the beginning of the `bibunit`.

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

```

315 \def\remequivalent#1\from#2{%
316   \let\given=#1%
317   \ifx#2\empty
318   \else
319     \edef#2{\expandafter\plugh#2\plugh}%
320   \fi
321 }
322 \def\plugh\do#1#2{%
323   \ifx#1\given
324   \else
325     \noexpand\do\noexpand#1%
326   \fi
327   \ifx#2\plugh
328   \hgulp\fi\plugh#2%
329 }
330 \def\hgulp\fi\plugh\plugh{\fi}

```

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

```

331 \remequivalent\bibtex\from\@preamblecmds

```

The command `\bibtex` calls `\@newlabel` 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 `\@newlabel` from the list of preamble command (like `\bibtex`), we redefine `\bibtex` to directly call `\@namedef` and issue no warning. We decided not to give even an info, because multiple cited references should be perfectly normal to bibunits. Similarly, we modify `natbib`’s `\bibtex`.

```

\bibtex
332 \AtBeginDocument{%
333   \ifpackageloaded{natbib}%
334     {\renewcommand\bibtex[2]{\global\@namedef{b@#1\@extra@binfo}{#2}}}%
335     {\renewcommand\bibtex[2]{\global\@namedef{b@#1}{#2}}}%
336 }

```

5.9 Package `babel` compatibility

`\input` If `babel` with options `frenchb` is loaded, some characters like the colon ‘:’ are made active. Since a colon frequently appears within a cite key, active characters have to be switched off for various commands. This can be done using `\bbl@redefine`.

First, check if `babel` is loaded, by checking if `\bbl@redefine` is defined. In this case, `\bbl@redefine` commands.

```

337 \AtBeginDocument{%
338   \ifundefined{bbl@redefine}%

```

```

339 {}%
340 {%
341   \bbl@redefine\@input#1{%
342     \@safe@activetrue\org@@input{#1}\@safe@activesfalse}%
343   \@ifpackageloaded{natbib}%
344   {%
345     \bbl@redefine\std@@citex[#1][#2]#3{%
346       \@safe@activetrue\org@std@@citex[#1][#2]{#3}\@safe@activesfalse}%
347     \bbl@redefine\bu@@citex[#1][#2]#3{%
348       \@safe@activetrue\org@bu@@citex[#1][#2]{#3}\@safe@activesfalse}%
349   }%
350   {
351     \@ifpackageloaded{jurabib}%
352     {%
353       \bbl@redefine\std@@citex[#1][#2]#3{%
354         \@safe@activetrue\org@std@@citex[#1][#2]{#3}\@safe@activesfalse}%
355       \bbl@redefine\bu@@citex[#1][#2]#3{%
356         \@safe@activetrue\org@bu@@citex[#1][#2]{#3}\@safe@activesfalse}%
357     }%
358     {% neither natbib nor jurabib loaded
359       \bbl@redefine\std@@citex[#1]#2{%
360         \@safe@activetrue\org@std@@citex[#1]{#2}\@safe@activesfalse}%
361     }%
362   }%
363 }%
364 }
365 </package>

```