The footbib package*

Eric Domenjoud
Eric.Domenjoud@loria.fr

2007/02/20

Contents

1 General overview

This package makes bibliographic references appear as footnotes. It defines a command `\footcite` which is similar to the `\cite` command of LaTeX but the references cited in this way are inserted at the bottom of the pages. This *foot*

*This file has version number 2.0.7, last revised 2007/02/20.

bibliography does not conflict with the standard one and both may exist simultaneously in a document. The command `\cite` may still be used to produce the standard bibliography.

The foot bibliography uses its own style and bibliographic database which are specified independently of the standard ones. Any standard bibliography style may be used. If the style does not provide explicit labels (e.g. `plain`), the references are numbered. The default is to number the references in the order in which they appear in the `thebibliography` environment. This may be overridden through options which allow the user to define a *numbering unit*. Then the references will be numbered in the order in which they are cited in the unit and the numbering restarts from 1 in each unit. The numbering unit may be a page, a double page, a chapter, a part or the whole document. Chapter and part may be used only if they are defined by the document class.

The user may also define a *citation unit* which may be a page, a double page, a chapter, a part or the whole document. The text of a reference will be inserted only once in each citation unit, on the page where the first citation occurs in the unit.

The mechanism used to put a reference only once in each citation unit may require several runs of LaTeX (usually at least two) before the references find their exact place. If necessary, LaTeX will issue, near the end of the document, a warning saying

```
Package footbib Warning: Bibliography not yet stable. Rerun LaTeX.
```

Using `footbib` in a document ⟨*doc*⟩`.tex` produces a file ⟨*doc*⟩`.fb.aux`. One must pass the argument ⟨*doc*⟩`.fb` to BibTeX to produce the bibliography which will be put in the file ⟨*doc*⟩`.fb.bbl`. The exact sequence of commands is

```
latex ⟨doc⟩
bibtex ⟨doc⟩.fb
latex ⟨doc⟩
latex ⟨doc⟩
...
```

Note: The name ⟨*doc*⟩`.fb.aux` might cause some problem on systems which do not allow a double extension in a file name or put a limit on the length of file names. A user command is provided to change it (see section 2.3).

At the beginning of the document, `footbib` inputs the bibliography from the file ⟨*doc*⟩`.fb.bbl` (or the name given by the user). If one wants to include the `thebibliography` environment in the main document, this may be done with a `filecontents` environment before the `\documentclass` command. See the LaTeX 2_ε documentation for more details about this environment.

2 User interface

2.1 Package options

2.1.1 oneside/twoside

`oneside` `twoside`

The `oneside` and `twoside` options affect the behaviour of `footbib` when either unit (citation or numbering) is the page. In `oneside` mode, the actual unit is a single page while in `twoside` mode, the unit is a double page. These options may be used to override a global `oneside` or `twoside` option.

2.1.2 citeonce[*]

`citeonce` `citeonce*`

The `citeonce` option overrides the default *citation unit*. `footbib` puts the text of a reference only once in each citation unit which may be a (double) page, a chapter, a part or the whole document. The default citation unit is the page in `oneside` mode and the double page in `twoside` mode. The new citation unit (`chapter`, `part` or `document`) is given as an optional argument between parentheses (`citeonce(chapter)`, `citeonce(part)` or `citeonce(document)`). If no argument is supplied, `document` is assumed. The argument `chapter` (resp. `part`) may be used only if the document class defines `\chapter` (resp. `\part`). The argument `page` may also be used but has a somehow special meaning. It defines a citation unit which is not overridden by another `citeonce` option but instead has a cumulative effect. For instance if one says

```
\usepackage[twoside,citeonce(page),citeonce(chapter)]{footbib}
```

then each double page and also each `\chapter` command starts a new citation unit. This may be useful if one wants a chapter to start a new unit even if it starts on a right page. It is only meaningful in `twoside` mode in conjunction with another `citeonce` option. In all other cases, it has no effect.

The `citeonce` option has a star form `citeonce*` with the same optional argument. When the star form is used, for each subsequent citation of a reference in the same citation unit but on another (double) page, the text of the reference is not omitted but replaced with a cross reference to the first citation in the same citation unit. The `page` argument is not available since it would have no effect.

2.1.3 firstcite

`firstcite`

The `firstcite` option affects the way the references are labelled. When the bibliography style does not provide explicit labels, the references are numbered. The default is to assign to each reference a *static* label which is its order in the `thebibliography` environment. The label is then the same for all citations of a given reference. The `firstcite` option causes the references to be numbered dynamically according to the order of their first citations. `firstcite` takes an optional argument between parentheses `firstcite(`⟨*unit*⟩`)` which defines the *numbering unit*. The numbering restarts then from 1 in each numbering unit. The argument ⟨*unit*⟩ may take the value `page`, `chapter`, `part` or `document`. If `page` is

used, then the numbering unit is a page in `oneside` mode and a double page in `twoside` mode. If no argument is supplied, `document` is assumed.

The effect of several `firstcite` options is cumulative in the sense that if one says for instance

```
\usepackage[twoside,firstcite(page),firstcite(chapter)]{footbib}
```

then each double page *and* each `\chapter` command starts a new numbering unit. This means that a `\chapter` command starts a new numbering unit even if it is on a right page.

If the bibliography style provides explicit labels, the `firstcite` option has no effect.

2.1.4 `crossrefs[*]` and `nocrossrefs`

`crossrefs` `crossrefs*` `nocrossrefs`

When an entry in the bibliographic database contains a `CROSSREF` field, BibTeX includes the cross-referenced entry in the bibliography and puts a `\cite` command in the entry where the `CROSSREF` field occurs. If no standard bibliography is produced, LaTeX will complain about an undefined reference. One may generally inhibit this behaviour of BibTeX by invoking it with the `-min-crossrefs=`⟨*number*⟩ option which tells how many times an entry must be cross-referenced before it is included in the bibliography and replaced with a `\cite` command. Setting ⟨*number*⟩ to a large value will generally inhibit the cross-referencing mechanism. However, this option has no effect if the cross-referenced entry is explicitely cited in the document.

The `crossrefs` option of `footbib` solves this problem by replacing each `\cite` command in a foot reference with `\footcite` (see the description of this command below). The star form `crossrefs*` replaces the `\cite` command with a `\footcite*`, which means that the text of the reference is not inserted. It is then the responsibility of the user to insert the text in the right place with a `\footnocite` command. Of course, standard citation through `\cite` is not possible anymore in a foot reference when either form of this option is used.

A `nocrossrefs` option is also provided to inhibit this behaviour in case it is not wanted but `crossrefs` occurs in the global options.

2.1.5 `split` and `nosplit`

`split` `nosplit`

The `nosplit` option tells `footbib` not to split the references across pages. The `split` option allows references to be split. `split` is the default and exists only to allow the user to override a global `nosplit` option.

2.2 Commands to generate the foot bibliography

`\footbibliography`

`\footbibliography{`⟨*file*⟩`,`⟨*file*⟩`,...}`

Defines the list of bibliographic databases for the foot bibliography. This command has the same syntax as the `\bibliography` command of LaTeX.

\footbibliographystyle `\footbibliographystyle{`⟨*style*⟩`}`
Defines the style of the foot bibliography. This command has the same syntax as the `\bibliographystyle` command of LaTeX.

\footcite `\footcite{`⟨*key*⟩`,`⟨*key*⟩`,...}`
Puts the list of labels in the text and the text of the references at the bottom of the page. The text of each reference is inserted at most once in a citation unit, even if it is cited several times.

\footcite* `\footcite*{`⟨*key*⟩`,`⟨*key*⟩`,...}`
Puts the list of labels in the text but does not put the reference at the bottom of the page.

\footnocite `\footnocite{`⟨*key*⟩`,`⟨*key*⟩`,...}`
Puts the reference at the bottom of the page but puts nothing in the text.

Note: The main purpose of the commands `\footcite*` and `\footnocite` is to solve the problem of a `\footcite` occuring inside an environment where the reference will be lost (for instance in a `minipage` or `tabular` environment, in a `\mbox`, etc.). In this case, if the reference is not cited otherwise on the same page, it won't show up at the bottom of the page. It suffices to add a `\footnocite` command just before or after this environment. The command `\footcite{`⟨*key*⟩`}` is more or less (but not completely) equivalent to `\footcite*{`⟨*key*⟩`}\footnocite{`⟨*key*⟩`}`.

2.3 Customisation

\footbibliographyname The basename of the `.aux` and `.bbl` files used for the foot bibliography may be redefined by

`\footbibliographyname{`⟨*name*⟩`}`

The default value is `\jobname.fb`[1] which causes `footbib` to read the bibliography from `\jobname.fb.bbl` and to use `\jobname.fb.aux` as an auxiliary file. This command may be used only in the preamble. The name supplied to `\footbibliographyname` must be different from the name of the main document.

\footcitelabel
\putfootcitelabel
\footcitelistformat The list of citations in the text may not be typeset in one step as done by the `\cite` command of LaTeX. The reason is that the command which creates the text of the reference must be inserted after each citation. The way (LA)TeX handles insertions makes them vanish if they occur in a box. Hence if the command which formats the list of citations puts them in a box, the text is lost and the references do not show up at the bottom of the page. All references could be inserted at once, either before or after the list of citations but if this list gets split across pages, the text of some references could show up on the wrong page. Hence the list is created one piece at a time and the text of the corresponding reference is inserted after each citation. The list of citation is created as follows:

[1] `\jobname` is a primitive TeX command which holds the name of the main document.

1) start of list
2) for each citation:
   a) if it is not the first one, separator of citations
   b) label of the reference, to which `\footcitelabel` is applied
   c) insertion of the text of the reference
3) end of list

The separator of citations is made of two parts: $\langle sep_1 \rangle$ and $\langle sep_2 \rangle$. The command `\putfootcitelabel` is applied to each component of the list, excepted $\langle sep_2 \rangle$ which is put as such. Typically, $\langle sep_2 \rangle$ is a separator which may disappear at a line break, like a penalty or some spacing. That's why `\putfootcitelabel` is not applied to it so that it won't be put in a box. The effect is as follows:

`\putfootcitelabel{`$\langle$*start of list*$\rangle$`}`
`\putfootcitelabel{\footcitelabel{`$\langle$*label* 1$\rangle$`}}`
$\langle$*insertion of the text of reference* 1$\rangle$
`\putfootcitelabel{`$\langle sep_1 \rangle$`}`
$\langle sep_2 \rangle$
`\putfootcitelabel{\footcitelabel{`label 2`}}`
$\langle$*insertion of the text of reference* 2$\rangle$
`\putfootcitelabel{`$\langle sep_1 \rangle$`}`
$\langle sep_2 \rangle$
⋮
`\putfootcitelabel{\footcitelabel{`label n`}}`
$\langle$*insertion of the text of reference* $n\rangle$
`\putfootcitelabel{`$\langle$*end of list*$\rangle$`}`

Each component of the list may be redefined as follows:

```
\renewcommand*\footcitelabel[1]{...}
\renewcommand*\putfootcitelabel[1]{...}
```
`\footcitelistformat`$\langle$*start of list*$\rangle\langle sep_1 \rangle\langle sep_2 \rangle\langle$*end of list*$\rangle$

Here are some examples of the variations allowed by this mechanism.

example 1: list of citations *a la* LaTeX: [label 1, label 2, ...]

```
\renewcommand*\footcitelabel[1]{#1}
\renewcommand*\putfootcitelabel[1]{#1}
\footcitelistformat[,{\penalty1000\␣}]
```

example 2: ditto but the list may not be cut

⋮

```
\footcitelistformat[,{\nobreak\␣}]
```

example 3: the list is raised and the labels are separated only by commas, without any space: [label 1,label 2,...]

```
\renewcommand*\footcitelabel[1]{#1}
\renewcommand*\putfootcitelabel[1]{\textsuperscript{\normalfont#1}}
\footcitelistformat[,{\penalty1000\relax}]
```

example 4: ditto, but no brackets around the list of labels: label 1,label 2,...

⋮

```
\footcitelistformat{},{\penalty1000\relax}{}
```

example 5: [label 1], [label 2], ...

```
\renewcommand*\footcitelabel[1]{[#1]}
\renewcommand*\putfootcitelabel[1]{#1}
\footcitelistformat{},{\penalty1000\␣}{}
```

The default definitions are the ones of example 3 above.

`\footbibskip` `\footbibrule` The foot bibliography is separated from the rest of the page by a vertical skip of length `\footbibskip` in which a horizontal line is drawn by the command `\footbibrule`. The height of the skip and the horizontal line may be redefined in the preamble by

```
\setlength\footbibskip{...}
\renewcommand\footbibrule{...}
```

CAUTION `\footbibrule` must take zero vertical space.

`\footreflabel` The label of the reference is formated by the macro `\footreflabel` which takes the label as argument. It may be redefined by `\renewcommand*\footreflabel[1]{...}`.

`\footrefstyle` The label and the text of the reference at the bottom of the page are typeset in the style defined by the command `\footrefstyle` which may be redefined in the preamble by `\renewcommad\footrefstyle{...}`. The default definition is `\normalfont\footnotesize`.

`\footxref` The options `citeonce*(`⟨*unit*⟩`)` tells `footbib` to replace the text of each reference but the first in each citation unit with a cross-reference to the last place where the full text of the reference appeared. The text of the cross-reference is generated by the command `\footxref` which takes two arguments: (1) the label and (2) the page of the last full citation. `\footxref` may be redefined in the preamble by

```
\renewcommand*\footxref[2]{...}.
```

3 Known and potential problems

- At present, the convergence is not proved. There is no guarantee that the references eventually find their place. However, `footbib` was used in large documents (several hundreds pages) and such a problem never occured.

- The foot bibliography is not sorted. The references appear at the bottom of the page in the order in which they are cited on the page.
- A `\footcite` command may not appear in a floating environment like `figure` or `table`.
- If a float is inserted at the bottom of the page, the foot bibliography is put *above* it, like footnotes.
- The result is not very nice in `twocolumn` mode. The references should be balanced between the two columns of the page (if there are two) or put in the right column like the package `ftnright` of Frank Mittelbach does for footnotes.
- `footbib` does not work with most packages which modify the output routine of LaTeX: `multicol`, `ftnright`, `floatflt`, `wrapfig`, etc.
- The references must not contain any `verbatim` environment. But `\verb` is allowed since it is sometime used to typeset filenames, URL's, etc.
- The braces must be balanced in the references, excepted the ones that might occur inside the argument of a `\verb` command. This implies that a reference may not contain say `\hbox\bgroup...}` which is otherwise correct in (LA)TeX.
- When references are numbered, the space between the label and the reference itself may be too large because the longest label is determined from the argument of `\begin{thebibliography}{`⟨*longest label*⟩`}` and its length is used for all references. If all references on a page have small numbers and the bibliography contains many references (say more than 100), this length is not reliable. The longest label should be deduced from the maximal number of references on a page, but this may not be known at the beginning of the document, at least at the first run. At the second run, the information could be deduces from what was written into the `.aux` file, provided the `\nofiles` command was not used (otherwise, the `.aux` file was not written). When per page numbering is used (option `firstcite(page)`), the longest label could also be simply initialised to `99` which is not too large and should be enough.

4 Implementation

4.1 Identification

```
1 ⟨*package⟩
2 \NeedsTeXFormat{LaTeX2e}
3 \ProvidesPackage{footbib}[\filedate\space v\fileversion\space(E.Domenjoud)]
```

4.2 Initial setup

Some badly behaved packages (written for LaTeX 2.09) change the catcodes before the beginning of the document and make some commands like `\@for` unusable.

The catcodes needed in the definitions are set here and restored at the end of the package.

```
\@makeother\‘
\edef\@tempa{\catcode\string‘\string‘\string=\the\catcode\string‘‘\relax}
\def\@tempb#1{\catcode‘\noexpand#1\string=\the\catcode‘#1\relax}
\edef\@tempa{\@tempa
  \@tempb\:\@tempb\?\@tempb\=\@tempb\<\@tempb\>\@tempb\+\@tempb\-%
  \@tempb\.\@tempb\’\@tempb\~}
\expandafter\AtEndOfPackage\expandafter{\@tempa}
\@makeother\: \@makeother\? \@makeother\= \@makeother\< \@makeother\>
\@makeother\+ \@makeother\- \@makeother\. \@makeother\’
\catcode‘\~=\active
```

4.3 Test of the output routine

If the LaTeX 2_ε format is more recent than the package, we test whether the output routine changed. If so, a warning is issued because the user might get unexpected results. The package should work with all previous versions of LaTeX 2_ε.

When `docstrip` is used to extract the package, this code is included only if the ‘`checkoutput`’ flag is used in addition to ‘`package`’.

```
⟨*checkoutput⟩
\@ifpackagelater{footbib}\fmtversion\@tempswafalse\@tempswatrue
\if@tempswa
\def\@tempa#1#2{\def\@tempb{#2}\ifx#1\@tempb\else\@tempswatrue\fi}
\@tempswafalse
\@tempa\@specialoutput{\ifnum\outputpenalty>-\@Mii\@doclearpage\else
  \ifnum\outputpenalty<-\@Miii\ifnum\outputpenalty<-\@MM\deadcycles\z@
  \fi\global\setbox\@holdpg\vbox{\unvbox\@cclv}\else\global\setbox
  \@holdpg\vbox{\unvbox\@holdpg\unvbox\@cclv\setbox\@tempboxa\lastbox
  \unskip}\@pagedp\dp\@holdpg\@pageht\ht\@holdpg\unvbox\@holdpg\@next
  \@currbox\@currlist{\ifnum\count\@currbox>\z@\advance\@pageht\@pagedp
  \ifvoid\footins\else\advance\@pageht\ht\footins\advance\@pageht\skip
  \footins\advance\@pageht\dp\footins\fi\ifvbox\@kludgeins\ifdim\wd
  \@kludgeins=\z@\advance\@pageht\ht\@kludgeins\fi\fi\@reinserts
  \@addtocurcol\else\@reinserts\@addmarginpar\fi}\@latexbug\ifnum
  \outputpenalty<\z@\if@nobreak\nobreak\else\addpenalty\interlinepenalty
  \fi\fi\fi\fi}
\@tempa\@doclearpage{\ifvoid\footins\ifvbox\@kludgeins{\setbox\@tempboxa
  \box\@kludgeins}\fi\setbox\@tempboxa\vsplit\@cclv to\z@
  \unvbox\@tempboxa\setbox\@tempboxa\box\@cclv\xdef\@deferlist{\@toplist
  \@botlist\@deferlist}\global\let\@toplist\@empty\global\let\@botlist
  \@empty\global\@colroom\@colht\ifx\@currlist\@empty\else\@latexerr
  {Float(s) lost}\@ehb\global\let\@currlist\@empty\fi\@makefcolumn
  \@deferlist\@whilesw\if@fcolmade\fi{\@opcol\@makefcolumn\@deferlist
  }\if@twocolumn\if@firstcolumn\xdef\@dbldeferlist{\@dbltoplist
  \@dbldeferlist}\global\let\@dbltoplist\@empty\global\@colht\textheight
  \begingroup\@dblfloatplacement\@makefcolumn\@dbldeferlist\@whilesw
  \if@fcolmade\fi{\@outputpage\@makefcolumn\@dbldeferlist}\endgroup\else
```

```
  \vbox{}\clearpage\fi\fi\else\setbox\@cclv\vbox{\box\@cclv\vfil}\@makecol
  \@opcol\clearpage\fi}
\@tempa\@makecol{\ifvoid\footins\setbox\@outputbox\box\@cclv\else\setbox
  \@outputbox\vbox{\boxmaxdepth\@maxdepth\unvbox\@cclv
  \vskip\skip\footins\color@begingroup\normalcolor\footnoterule\unvbox
  \footins\color@endgroup}\fi\let\@elt\relax\xdef\@freelist{\@freelist
  \@midlist}\global\let\@midlist\@empty\@combinefloats\ifvbox\@kludgeins
  \@makespecialcolbox\else\setbox\@outputbox\vbox to\@colht{\@texttop\dimen@
  \dp\@outputbox\unvbox\@outputbox\vskip-\dimen@\@textbottom}\fi\global
  \maxdepth\@maxdepth}
\@tempa\@reinserts{\ifvoid\footins\else\insert\footins{\unvbox\footins}\fi
  \ifvbox\@kludgeins\insert\@kludgeins{\unvbox\@kludgeins}\fi}
\fi
\if@tempswa
  \PackageError{footbib}{the output routine of LaTeX changed}
  {The output routine of LaTeX changed since the current version of
   `footbib'.\MessageBreak Since `footbib' patches this routine,
   using it may produce unexpected\MessageBreak results. Send a mail to \space
   Eric.Domenjoud@loria.fr \space to get a new version.\MessageBreak\MessageBreak
   Type \space X <return> \space to quit or cross your fingers and
   just type <return>.\MessageBreak}
\fi
```
⟨/checkoutput⟩

4.4 Package Options

4.4.1 Initial code

\iffb@twoside
\iffb@citeonce
\iffb@pagecite
\fb@chaptercite
\fb@partcite
\iffb@firstcite
\iffb@pagenum
\fb@chapternum
\fb@partnum
\iffb@xref
\iffb@crossrefs
\iffb@xcrossrefs
\iffb@nosplit

First we define some switches which record the user options. The switch `\iffb@twoside` is initialised from the current value of `\if@twoside` because default global options are not passed to packages. If one says `\documentclass{book}` then the document is in `twoside` mode but the packages *don't know* it.

```
\newif\iffb@twoside     \let\iffb@twoside\if@twoside
\newif\iffb@citeonce    \fb@citeoncefalse
\newif\iffb@pagecite    \fb@pagecitefalse
\let\fb@chaptercite\@empty
\let\fb@partcite\@empty
\newif\iffb@firstcite   \fb@firstcitefalse
\newif\iffb@pagenum     \fb@pagenumfalse
\let\fb@chapternum\@empty
\let\fb@partnum\@empty
\newif\iffb@xref        \fb@xreffalse
\newif\iffb@crossrefs   \fb@crossrefsfalse
\newif\iffb@xcrossrefs  \fb@xcrossrefsfalse
\newif\iffb@nosplit     \fb@nosplitfalse
```

\fb@checksec

The macro `\fb@checksec` checks whether its first argument (a sectioning command) is defined. If so, the second argument (a list of command) is executed. Otherwise an error is raised and the second argument is discarded. It is called

while processing the options which must patch a sectioning command.

```
\newcommand\fb@checksec[2]{%
  \ifx#1\@undefined
    \PackageError{footbib}{Bad option `\CurrentOption'}%
    {{footbib}: The current document class does not define `\string#1'}%
  \else
    #2%
  \fi}
```

4.4.2 Declaration and processing of options

oneside, twoside

```
\DeclareOption{oneside}{\fb@twosidefalse}
\DeclareOption{twoside}{\fb@twosidetrue}
```

firstcite

```
\DeclareOption{firstcite}{\fb@firstcitetrue}
\DeclareOption{firstcite(page)}{\fb@firstcitetrue\fb@pagenumtrue}
\DeclareOption{firstcite(chapter)}{\fb@firstcitetrue
    \fb@checksec\chapter{\def\fb@chapternum{\fb@newnumunit}}}
\DeclareOption{firstcite(part)}{\fb@firstcitetrue
    \fb@checksec\part{\def\fb@partnum{\fb@newnumunit}}}
\DeclareOption{firstcite(document)}{\fb@firstcitetrue}
```

citeonce, citeonce*

```
\DeclareOption{citeonce}{\fb@citeoncetrue}
\DeclareOption{citeonce(page)}{\fb@pagecitetrue}
\DeclareOption{citeonce(chapter)}{\fb@citeoncetrue
    \fb@checksec\chapter{\def\fb@chaptercite{\fb@newciteunit}}}
\DeclareOption{citeonce(part)}{\fb@citeoncetrue
    \fb@checksec\part{\def\fb@partcite{\fb@newciteunit}}}
\DeclareOption{citeonce(document)}{\fb@citeoncetrue}
\DeclareOption{citeonce*}{\fb@citeoncetrue\fb@xreftrue}
\DeclareOption{citeonce*(chapter)}{%
    \ExecuteOptions{citeonce(chapter)}\fb@xreftrue}
\DeclareOption{citeonce*(part)}{%
    \ExecuteOptions{citeonce(part)}\fb@xreftrue}
\DeclareOption{citeonce*(document)}{\ExecuteOptions{citeonce*}}
```

crossrefs, crossrefs*, nocrossrefs

```
\DeclareOption{crossrefs}{\fb@crossrefstrue\fb@xcrossrefstrue}
\DeclareOption{crossrefs*}{\fb@crossrefstrue\fb@xcrossrefsfalse}
\DeclareOption{nocrossrefs}{\fb@crossrefsfalse\fb@xcrossrefsfalse}
```

split, nosplit

```
\DeclareOption{split}{\fb@nosplitfalse}
\DeclareOption{nosplit}{\fb@nosplittrue}
```

```
\ProcessOptions*
\iffb@pagecite
  \fb@citeoncefalse
  \fb@xreffalse
\fi
\let\fb@firstcitetrue\@undefined  \let\fb@firstcitefalse\@undefined
\let\fb@citeoncetrue\@undefined   \let\fb@citeoncefalse\@undefined
```

```
\let\fb@pagecitetrue\@undefined   \let\fb@pagecitefalse\@undefined
\let\fb@xreftrue\@undefined       \let\fb@xreffalse\@undefined
\let\fb@crossrefstrue\@undefined  \let\fb@crossrefsfalse\@undefined
\let\fb@xcrossrefstrue\@undefined \let\fb@xcrossrefsfalse\@undefined
\let\fb@pagenumtrue\@undefined    \let\fb@pagenumfalse\@undefined
\let\fb@nosplittrue\@undefined    \let\fb@nosplitfalse\@undefined
\let\fb@checksec\@undefined
```

\chapter
\part

At the beginning of the document, the commands `\chapter` and `\part` are patched if necessary so that they start a new citation or numbering unit. This is achieved by adding in front of them the commands hold in `\fb@chaptercite`, `\fb@chapternum`, `\fb@partcite` and `\fb@partnum` defined while processing the options. When a command is patched, a `\clearpage` is added so that a new unit always starts at the top of a page.

```
\AtBeginDocument{%
  \begingroup
    \def\@tempb#1{%
      \ifx\@tempa\@empty\else
        \edef\@tempa{\noexpand\clearpage\@tempa}%
        \toks@\expandafter\expandafter\expandafter{\expandafter\@tempa#1}%
        \xdef#1{\the\toks@}%
      \fi}%
    \let\fb@newnumunit\relax
    \let\fb@newciteunit\relax
    \edef\@tempa{\fb@partcite\fb@partnum}%
    \@tempb\part
    \edef\@tempa{\fb@chaptercite\fb@chapternum}%
    \@tempb\chapter
  \endgroup}
```

4.5 Customisation

4.5.1 Basename of the files used for the foot bibliography

\footbibliographyname

The basename of the `.aux` and `.bbl` files is produced by the command `\fb@bibname` which is redefined by a call to `\footbibliographyname` in the preamble.

`\footbibliographyname` first checks that its argument is different from `\jobname`. Since the characters in `\jobname` have catcode 12 (other), the first two commands below yield the argument of `\footbibliographyname` also with catcodes 12 so that it may be compared to `\jobname`.

```
\newcommand*\footbibliographyname[1]{%
  \edef\@tempa{#1}%
  \edef\@tempa{\expandafter\strip@prefix\meaning\@tempa}%
  \edef\@tempb{\jobname}%
  \ifx\@tempa\@tempb
    \PackageError{footbib}{Bad argument `#1'\on@line}%
      {The name supplied to `\string\footbibliographyname' must be
         different from the name\MessageBreak
       of the current document to avoid conflicts with the standard
```

```
150          bibliography.\MessageBreak\MessageBreak
151        Type \space X <return> \space to quit.\MessageBreak}%
152   \else
153     \xdef\fb@bibname{#1}%
154   \fi}
155 \@onlypreamble\footbibliographyname
156 \footbibliographyname{\jobname.fb}
```

4.5.2 Layout of the list of citations in the text

\footcitelabel
\putfootcitelabel
\footcitelistformat
\fb@putfootcitelabel
\fb@citestart
\fb@citesep
\fb@citeend

Here we define the layout parameters for the list of citations in the text. We define an additional macro `\fb@putfootcitelabel` which is essentially `\putfootcitelabel`. Only `\scriptspace` and `\mathsurround` are set to 0 pt in case `\putfootcitelabel` involves some math. The modified version is applied to all components of the citation list but `\fb@citeend` to which the standard version is applied. This avoids unwanted spacing inside the list while allowing some additional spacing after it. The macro `\footcitelistformat` defines the macros `\fb@citestart`, `\fb@citesep` and `\fb@citeend` used by `\fb@cite` to build the list of citations in the text.

```
157 \newcommand*\footcitelabel[1]{#1}
158 \newcommand*\putfootcitelabel[1]{\textsuperscript{\normalfont#1}}
159 \newcommand*\fb@putfootcitelabel[1]{%
160   {\m@th\scriptspace\z@\putfootcitelabel{#1}}}
161 \newcommand*\footcitelistformat[4]{%
162   \def\fb@citestart{\fb@putfootcitelabel{#1}}%
163   \def\fb@citesep{\fb@putfootcitelabel{#2}#3}%
164   \def\fb@citeend{\putfootcitelabel{#4}}}%
165 \footcitelistformat[,{\penalty\@m}]
```

4.5.3 Style of the foot bibliography

\footbibskip
\footbibrule
\footrefstyle
\footreflabel
\footxref
\fb@ins

Here are defined all the layout parameters for the foot bibliography. `\fb@ins` is the insertion number for the foot bibliography. It is not really the right place for its declaration but it is needed to define the user definable parameter `\footbibskip`. By the way we set all the parameters for these insertions: 1 to 1 magnification and no limit on the height of the foot bibliography.

```
166 \newinsert\fb@ins
167 \count\fb@ins=1000
168 \AtBeginDocument{\dimen\fb@ins=\textheight}
169 \newcommand\footbibskip{\skip\fb@ins}
170 \footbibskip=\bigskipamount
171 \newcommand\footbibrule{\kern-3\p@\hrule\kern 2.6\p@}% \hrule is .4pt high
172 \newcommand\footrefstyle{\normalfont\footnotesize}
173 \newcommand*\footreflabel[1]{[#1]}
174 \newcommand*\footxref[2]{See~#1~on page~#2.}
```

4.6 Some useful definitions

`\fb@vedef` `\fb@doactive`

The keys of the references must be read and written *more or less* verbatim. Since some packages make some characters permanently active, the catcodes should be changed before reading or writing a key or a list of keys and reset afterward. Unfortunately, in some situations this is not possible because the catcodes have already been attached to the characters. One solution is to scan the key and replace each active character with its non-active equivalent but this is fairly costly and does not work if some active character have been let equal to a non-active one or is hidden in a command occuring in the (list of) key(s). Another solution is to use `\meaning` to get a *verbatim* copy of the keys but this does not allow them to contain commands like in `\foocite{\mylistofcitations}` The method used here consists in redefining the active characters so that their expansion produces the same character with a catcode 12 (other). Then an expansion of the key yields an almost *verbatim* copy of it. Only then, `\meaning` is used to *remove* the category code of any special character which might remain in the key like `$` or `_`. This mechanism allows the list of keys to contain itself commands which expand in the normal way. All characters which may both be active in the document and appear in the key of a reference must be treated in this way. Since BibTeX allows almost any character in a key, it is safer to consider all characters which may be active in a document. No matter that they indeed are. The command `\dospecials` usually contains, among other, all such characters, each one escaped and preceded by `\do`. The LaTeX 2$_\varepsilon$ kernel defines `\dospecials` as

```
\def\dospecials{\do\␣\do\\\do\{\do\}\do\$\do\&\do\#\do\^\do\_\do\%\do\~}
```

Any package which define new *special* characters should add them to this list.

An active character is needed to start with. Since `~` was made active at the beginning of the package, it may be safely used for this purpose.

```
\newcommand*\fb@doactive[1]{\lccode`~=`#1\lowercase{\def~{\string~}}}
\newcommand*\fb@vedef[2]{%
  \begingroup
    \let\do\fb@doactive \dospecials
    \edef\@tempa{\endgroup\def\noexpand#1{#2}}%
  \@tempa
  \edef#1{\expandafter\strip@prefix\meaning#1}}
```

`\fb@namexdef`

`\fb@namexdef` is similar to the `\@namedef` command of LaTeX but uses `\xdef` instead of `\def`

```
\newcommand*\fb@namexdef[1]{\expandafter\xdef\csname#1\endcsname}
```

`\fb@auxout` `\fb@writeaux`

`\fb@auxout` is the auxiliary file used to record information about citations and as input to BibTeX. The commands `\footbibliographystyle` and `\footbibliography` do an immediate `\write` to this file. However, since these commands may be used in the preamble, this file might not yet be open for writing at the time they are used. We must wait until the end of the preamble before opening the auxiliary file to give the user a chance to define its name with `\footbibliographyname`.

Therefore, we define the macro `\fb@writeaux` which postpones the write until the beginning of the document. It is somehow a *delayed immediate write* which means that the write will be performed as soon as possible, i.e. as soon as the auxiliary file is open for writing. This definition is temporary and will be changed to a *true* immediate write by `\AtBeginDocument`.

```
\newwrite\fb@auxout
\newcommand*\fb@writeaux[2]{%
  \AtBeginDocument{%
    \if@filesw
      \immediate\write\fb@auxout{\string#1{#2}}%
    \fi}}
```

4.7 Units handling

`\fb@numunit` `\fb@citeunit` `\fb@xrefunit`

We handle 3 counters: the *numbering unit* counter (`\fb@numunit`), the *citation unit* counter (`\fb@citeunit`) and the *cross-referencing unit* counter (`\fb@xrefunit`). This last counter is meaningful only if a `citeonce*` option was used. It essentially counts pages in `oneside` mode and double pages in `twoside` mode. However, if the citation unit is a chapter (resp. a part), each `\chapter` (resp. `\part`) command also increments this counter.

```
\newcount\fb@numunit  \fb@numunit\@ne
\newcount\fb@citeunit \fb@citeunit\@ne
\newcount\fb@xrefunit \fb@xrefunit\@ne
```

`\fb@refciteunit` `\fb@refxrefunit` `\fb@refnumunit` `\fb@refpage` `\fb@theunits` `\fb@getunits` `\fb@get@units`

The macros `\fb@ref...unit` and `\fb@refpage` hold the units and the page of the current citation.

```
\newcommand\fb@theunits{%
  {\the\fb@citeunit}{\the\fb@xrefunit}{\the\fb@numunit}{\thepage}}
\newcommand*\fb@getunits[1]{%
  \ifx#1\relax
    \xdef#1{{\fb@key}\fb@theunits}%
  \fi
  \expandafter\fb@get@units#1}
\newcommand*\fb@get@units[5]{%
  \def\fb@refciteunit{#2}%
  \def\fb@refxrefunit{#3}%
  \def\fb@refnumunit{#4}%
  \def\fb@refpage{#5}}%
```

`\fb@newciteunit` `\fb@newnumunit`

The macros `\fb@newciteunit` and `\fb@newnumunit` are called by the patched versions of `\part` or `\chapter` to start a new citation or numbering unit. The patched sectioning unit also forces a page break so that a unit always starts at the top of a page.

```
\newcommand\fb@newciteunit{%
  \global\advance\fb@citeunit\@ne
  \global\advance\fb@xrefunit\@ne}
\newcommand\fb@newnumunit{\global\advance\fb@numunit\@ne}
```

\fb@checkpage
\fb@prevpage

Each time the page counter is incremented, i.e. at the top of a new page, the macro \fb@checkpage updates the units counters if necessary. We first check whether the current page is a right page. In oneside mode, it is never the case. In twoside mode, it is the case if its number is \fb@prevpage + 1 and is odd, where \fb@prevpage is a counter which holds the number of the last shiped out page. If the number of the current page is not \fb@prevpage + 1, it means that either the user has manually changed the page counter or the page numbering has changed. In both cases, we consider the current page as a left page. If the current page is a right page, we do nothing. Otherwise, it may start a new unit[2] and we update the unit counters.

```
\newcount\fb@prevpage \fb@prevpage\@ne
\newcommand*\fb@checkpage{%
  \@tempswatrue
  \iffb@twoside
    \global\advance\fb@prevpage\@ne
    \ifnum\fb@prevpage=\c@page
      \ifodd\c@page
        \@tempswafalse
      \fi
    \fi
  \fi
  \if@tempswa
```

The counter \fb@numunit is incremented if the switch \iffb@pagenum is true, i.e. the option firstcite(page) was used.

```
    \iffb@pagenum
      \global\advance\fb@numunit\@ne
    \fi
```

The switch \iffb@citeonce is true iff a citeonce or citeonce* option was used. In this case, the \fb@citeunit counter is incremented by the \chapter or \part command. We just increment \fb@xrefunit in case citeonce* was used. If \iffb@citeonce is false, the citation unit is the (double) page and we increment \fb@citeunit. Since no citeonce* option was used, we do not need to handle \fb@xrefunit.

```
    \iffb@citeonce
      \global\advance\fb@xrefunit\@ne
    \else
      \global\advance\fb@citeunit\@ne
    \fi
  \fi
  \global\fb@prevpage\c@page}
```

\c@fb%checkpage

The next definition is a trick to make the macro \fb@checkpage execute each time the page counter is incremented. Each counter ⟨*cnt*⟩ declared by \@addtoreset{⟨*cnt*⟩}{page} is reset to 0 by \global\c@⟨*cnt*⟩\z@ when the

[2] A right page may actually also start a new unit if for instance the firstcite(chapter) option is in effect and the current page starts a new chapter. But in this case, the units are updated by the \chapter command.

counter `page` is incremented. We define a macro `\c@`*xxx* which *looks like* a counter and we say `\@addtoreset{`*xxx*`}{page}`. To be sure that the user will never define a counter named *xxx*, we name our macro `\c@fb%checkpage`. The name of the associated *pseudo counter* is `fb%checkpage` that the user may normally not type. Each time the page counter is incremented, `\global\c@fb%checkpage\z@` is executed. The macro `\c@fb%checkpage` starts with an assignment (`\count@\z@`) which *uses* the `\global` and ends with a counter (`\count@`) which *gobbles* the following `\z@`. Since we change the catcode of `%`, it may not be used for comments below.

```
\catcode`\%=11
\newcommand\c@fb%checkpage{\count@\z@
  \fb@checkpage
  \count@}
\@addtoreset{fb%checkpage}{page}
\catcode`\%=14
```

4.8 Commands to handle the references

`\fb@refcount` The counter `\fb@refcount` holds the number of the last numbered reference. It is reset to 0 at the beginning of each numbering unit.

```
\newcount\fb@refcount
```

`\fb@lbl` `\fb@txt` The token registers `\fb@lbl` and `\fb@txt` always holds the label and the text of the current reference.

```
\newtoks\fb@lbl
\newtoks\fb@txt
```

`\fb@setref` `\fb@getref` `\fb@r.`⟨*key*⟩ `\fb@setref` stores the current value of the token registers `\fb@lbl` and `\fb@txt` in the macro `\fb@r.`⟨*key*⟩ where ⟨*key*⟩ is the key of the current reference. This key is always stored in the macro `\fb@key`. `\fb@setref` is called each time a component of a reference changes: when it is first read at the beginning of the document, and when the dynamic label or the text of the reference has been updated. `\fb@getref` does the converse: given a key, it updates `\fb@lbl` and `\fb@txt` from `\fb@r.`⟨*key*⟩.

```
\newcommand\fb@setref{%
  \fb@namexdef{fb@r.\fb@key}{{\the\fb@lbl}{\the\fb@txt}}}
\newcommand\fb@getref{\afterassignment\fb@txt\fb@lbl}
```

`\fb@setlbl` `\fb@getlbl` `\fb@setlbl` updates the dynamic label of a reference. `\fb@getlbl` gets the label after calling `\fb@setlbl` if necessary to update it.

```
\newcommand\fb@setlbl{%
  \global\advance\fb@refcount\@ne
  \edef\@tempa{{\fb@refnumunit}{\the\fb@refcount}}%
  \global\fb@lbl\expandafter{\expandafter\fb@getlbl\@tempa}%
  \fb@setref
  \the\fb@refcount}
\newcommand*\fb@getlbl[2]{\ifnum\fb@refnumunit=#1{#2}\else\fb@setlbl\fi}
```

\fb@settxt
\fb@gettxt

\fb@settxt and \fb@gettxt are similar to \fb@setlbl and \fb@getlbl but update the text of the reference instead of the label. When \fb@settxt is called, the label must have been expanded just before so that it has been updated if necessary. Then \fb@lbl has either the form {⟨*static label*⟩} or the form \fb@getlbl{⟨*current numunit*⟩}{⟨*dynamic label*⟩}. In the later case, the test at the beginning of \fb@getlbl must succeed so that in both cases, the first command in \fb@settxt below assigns the label to \toks@.

```
\newcommand\fb@settxt[1]{%
  \toks@=\the\fb@lbl
  \toks@\expandafter\expandafter\expandafter{\expandafter
    \footreflabel\expandafter{\the\toks@}}%
  \edef\@tempa{{\fb@refciteunit}{\the\toks@}{\fb@refpage}}%
  \global\fb@txt\expandafter{\expandafter\fb@gettxt\@tempa{#1}}%
  \fb@setref
  #1}
\newcommand\fb@gettxt[4]{%
  \ifnum\fb@refciteunit=#1\relax
    \footxref{#2}{#3}%
  \else
    \fb@settxt{#4}%
  \fi}
```

4.9 Commands to handle the foot bibliography

\footbibliography
\footbibliographystyle

\footbibliography and \footbibliographystyle are the commands which define the bibliography file and the bibliography style. They just write their argument to the auxiliary file. If they are used in the preamble, the \fb@writeaux macro in use is the delayed one. The write will actually take place at the beginning of the document.

```
\newcommand\footbibliography{\fb@writeaux\bibdata}
\newcommand\footbibliographystyle{\fb@writeaux\bibstyle}
```

\fb@refnotfound

When a reference is not found by LaTeX, it is replaced by a default one generated by the command \fb@refnotfound which takes the key of the reference as an argument and expands to the default label and the default text.

```
\newcommand*\fb@refnotfound[1]{%
  {?}{*** ERROR: citation `{\normalfont\ttfamily\bfseries#1}' undefined ***}}
```

4.9.1 Creation of the foot references

\footcite
\footcite*
\footnocite
\fb@@cite
\fb@optlbl
\iffb@lbl
\iffb@txt

All 3 citation commands \footcite, \footcite* and \footnocite actually call the same macro \fb@cite. Before this call, they just set the flags \iffb@lbl and \iffb@txt according to whether the label and the text of the reference are requested. The settings are as follows:

	\iffb@lbl	\iffb@txt
\footcite	true	true
\footcite*	true	false
\footnocite	false	true

In addition, `\footcite` calls `\fb@@cite` to get the optional argument which is put in `\fb@optlbl`.

```
\newif\iffb@lbl
\newif\iffb@txt
\DeclareRobustCommand\footcite{%
  \fb@lbltrue\@ifstar{\fb@txtfalse\fb@@cite}{\fb@txttrue\fb@@cite}}
\newcommand\footnocite{\fb@lblfalse\fb@txttrue\fb@cite}
\newcommand*\fb@@cite[1][\@nil]{%
  \def\fb@optlbl{#1}%
  \ifx\fb@optlbl\@nnil
    \let\fb@optlbl\relax
  \else
    \def\fb@optlbl{\fb@putfootcitelabel{, #1}}%
  \fi
  \fb@cite}
```

`\fb@cite` `\fb@cite` is the macro which handles the list of citations. It calls `\fb@xcite` to produce the actual label and insert the text of each individual reference.

```
\newcommand*\fb@cite[1]{%
  \fb@vedef\fb@keys{#1}%
  \iffb@lbl
    \fb@citestart
    \def\fb@citea{\let\fb@citea\fb@citesep}%
  \fi
```

If the list of citation is empty, the `\@for` loop is not entered and no warning is issued. Therefore, we check for this now.

```
  \ifx\fb@keys\@empty
    \PackageWarning{footbib}{Empty citation on page \thepage}%
  \fi
  \@for\fb@key:=\fb@keys\do{%
    \iffb@lbl\fb@citea\fi
    \fb@xcite
    \ifx\fb@deferredcite\@empty\else
      \begingroup
        \fb@lblfalse
        \expandafter\fb@xnocite\fb@deferredcite\@nil
      \endgroup
    \fi}%
  \iffb@lbl\fb@optlbl\fb@citeend\fi}
```

`\fb@xfootcite` `\fb@deferredcite` The macro `\fb@xfootcite` is a replacement for the `\cite` command of LaTeX inside a foot reference if the `crossrefs` or `crossrefs*` option was used. In both cases, `\fb@xfootcite` performs a `\footcite*`. If the `crossrefs` option was used, in addition, `\fb@xfootcite` adds globally the list of citation keys to

the list \fb@deferredcite. After the insertion of the current reference has been completed, a \footnocite will be performed for each key in the list. The format of this list is ⟨*key*⟩,...,⟨*key*⟩, (the trailing comma makes it easier to handle than ⟨*key*⟩,...,⟨*key*⟩ and allows to distinguish between an empty list and a list containing only an empty element). This list is initially empty.

```
\newcommand\fb@xfootcite[2][\@nil]{%
  \footcite*[#1]{#2}%
  \iffb@xcrossrefs
    \fb@vedef\fb@keys{#2}%
    \xdef\fb@deferredcite{\fb@deferredcite\fb@keys,}%
  \fi}
\let\fb@deferredcite\@empty
```

\fb@xnocite After the insertion of the current reference has been completed, if some deferred cross-references are present, the macro \fb@xnocite is called. It calls \fb@xcite with \iffb@lbl=*false* to insert the text of the cross-references if necessary. This might produce more deferred cross-references which will be added to \fb@deferredcite.

```
\def\fb@xnocite#1,#2\@nil{%
  \gdef\fb@deferredcite{#2}%
  \def\fb@key{#1}%
  \fb@xcite
  \ifx\fb@deferredcite\@empty
    \let\@tempa\@gobble
  \else
    \let\@tempa\fb@xnocite
  \fi
  \expandafter\@tempa\fb@deferredcite\@nil}
```

\fb@xcite The macro \fb@xcite is called both by \fb@cite and \fb@xnocite to handle each individual citation. It writes to the auxiliary file the information about the citation, puts the label in the text if requested and put the text of the reference on the page if necessary. If the reference is not found, it issues a warning.

The first command in \fb@xcite removes any space in front of the key. \@empty is inserted after the key to prevent an error in case it is empty.

If the key is empty or the reference is undefined, the LaTeX command \G@refundefinedtrue is used to set the switch \if@refundefined which indicates that some reference was undefined.

```
\newcommand\fb@xcite{%
  \edef\fb@key{\expandafter\@firstofone\fb@key\@empty}%
  \ifx\fb@key\@empty
    \PackageWarning{footbib}{Empty citation on page \thepage}%
    \G@refundefinedtrue
  \else
    \@ifundefined{fb@r.\fb@key}%
      {\G@refundefinedtrue
       \PackageWarning{footbib}{%
          Citation `\fb@key' on page \thepage \space undefined}%
```

```
      \fb@getref\fb@refnotfound\fb@key\fb@setref}%
     {\fb@getref\@nameuse{fb@r.\fb@key}}%
   \fb@bibcite
   \iffb@lbl\fb@putfootcitelabel{\footcitelabel{\the\fb@lbl}}\fi
```

The text of the reference is inserted if requested (`\iffb@txt`=*true*) and either it has not yet been inserted in the current citation unit, or a `citeonce*` option was used and the last citation was on another (double) page.

```
  \iffb@txt
    \expandafter\let\expandafter\@tempa\csname
        fb@fn\fb@key.\fb@refciteunit\endcsname
    \@tempswatrue
    \ifx\@tempa\relax\else
      \iffb@xref
        \ifx\@tempa\fb@refxrefunit
          \@tempswafalse
        \fi
      \else
        \@tempswafalse
      \fi
    \fi
    \if@tempswa
      \fb@namexdef{fb@fn\fb@key.\fb@refciteunit}{\fb@refxrefunit}%
      \fb@citefn
    \fi
  \fi
\fi}
```

`\fb@bibcite` `\fb@lastrefnumunit` `\fb@citecount`

The command `\fb@bibcite` writes to the auxiliary file all the informations about the current citation: the key, the units, and the page. It also resets to 0 the counter `\fb@refcount` if the numbering unit changed between the last reference and the current one. This counter is used to number the references. The counter `\fb@lastrefnumunit` holds the numbering unit of the last reference. The counter `\fb@citecount` holds the number of the current citation.

```
\newcount\fb@citecount
\newcount\fb@lastrefnumunit \fb@lastrefnumunit\m@ne
\newcommand\fb@bibcite{%
  \fb@writeaux\citation\fb@key
  \global\advance\fb@citecount\@ne
  \expandafter\fb@getunits\csname fb@c.\the\fb@citecount\endcsname
  \ifnum\fb@refnumunit=\fb@lastrefnumunit\else
    \global\fb@lastrefnumunit\fb@refnumunit\relax
    \global\fb@refcount\z@
  \fi
  \if@filesw
    \write\fb@auxout\expandafter{\expandafter\string\expandafter
      \bibcite\expandafter{\fb@key}\fb@theunits}%
  \fi}
```

\fb@citefn

\fb@citefn inserts the text of the reference. It is called only if the reference was not already cited in the same citation unit or a citeonce* option was used. The code is mainly borrowed from the footnotes handling in LaTeX.

```
\newcommand\fb@citefn{%
  \insert\fb@ins{%
    \reset@font\footrefstyle
    \interlinepenalty\iffb@nosplit\@M\else\interfootnotelinepenalty\fi
    \splittopskip 1.2\ht\strutbox
    \splitmaxdepth \dp\strutbox
    \floatingpenalty \@MM
    \hsize\columnwidth
    \@parboxrestore
    \ifx\newblock\@undefined\let\newblock\relax\fi
    \iffb@crossrefs\let\cite\fb@xfootcite\fi
    \@tempdima\fb@lblwidth
    \advance\@tempdima\labelsep
    \leftskip\@tempdima
    \color@begingroup
      \setbox\@tempboxa\hbox{\footreflabel{\the\fb@lbl}}%
      \hskip-\@tempdima
      \ifdim\wd\@tempboxa<\fb@lblwidth
        \hb@xt@\fb@lblwidth{\unhbox\@tempboxa\hfil}%
      \else
        \box\@tempboxa
      \fi
      \hskip\labelsep
      \rule\z@{1.2\ht\strutbox}\ignorespaces\the\fb@txt\@finalstrut\strutbox
    \color@endgroup}}
```

4.10 AtBeginDocument, AtEndDocument

At the beginning of the document, we read the bibliography file and record all the references. This is memory consuming but the only alternative is to read again the bibliography file for each \footcite command which would be much slower. We must wait until the beginning of the document to give the user a chance to redefine \fb@bibname through \footbibliographyname. The preamble of the bibliography is executed once for all when the bibliography is read. After recording the bibliography, we read the .aux file (if it exists) and record the parameters (citation and numbering unit, page, etc.) of all citations as determined during the previous run. The parameters of the n^{th} citation are recorded in the macro \fb@c.$\langle n \rangle$.

```
\AtBeginDocument{%
```

Save the current value of \thebibliography and redefine it

```
  \let\fb@savethebibliography\thebibliography
  \let\thebibliography\fb@thebibliography
  \let\fb@thebibliography\@undefined
```

Read the `bbl` file. This executes the preamble, and if a `thebibliography` environment is found, sets `\fb@lblwidth` to the length of the longest label and records all references. `\fb@lblwidth` is initialised with a negative value which allows us to detect afterward whether a `thebibliography` environment was present.

```
\global\fb@lblwidth=-\maxdimen
\fb@refcount\z@
\@input{\fb@bibname.bbl}%
```

Restore the standard value of `\thebibliography`

```
\let\thebibliography\fb@savethebibliography
\let\fb@savethebibliography\@undefined
```

Assign a default value to `\fb@lblwidth` in case the `bbl` file was not found or contained no `thebibliography` environment.

```
\ifdim\fb@lblwidth<\z@
  \settowidth\fb@lblwidth
    {\footrefstyle\footreflabel{\expandafter\@firstoftwo\fb@refnotfound{?}}}%
\fi
```

`\bibcite` `\fb@c.⟨num⟩`

Then read the auxiliary file and record the parameters of each citation.

```
\fb@citecount\z@
\begingroup
  \let\citation\@gobble \let\bibstyle\@gobble \let\bibdata\@gobble
  \def\bibcite#1#2#3#4#5{%
      \advance\fb@citecount\@ne
      \fb@vedef\fb@key{#1}%
      \fb@namexdef{fb@c.\the\fb@citecount}{{\fb@key}{#2}{#3}{#4}{#5}}}%
  \@input{\fb@bibname.aux}%
\endgroup
```

`\fb@auxout`

Prepare the auxiliary file for writing

```
\if@filesw
  \immediate\openout\fb@auxout=\fb@bibname.aux
  \immediate\write\fb@auxout{\relax}%
\fi
```

`\fb@writeaux`

Finally, define a non delayed version of `\fb@writeaux`.

```
\renewcommand*\fb@writeaux[2]{%
  \if@filesw
    \immediate\write\fb@auxout{\string#1{#2}}%
  \fi}%
}
```

`\fb@thebibliography` `\fb@lblwidth`

The macro `\fb@thebibliography` records the length of the longest label in the ⟨*dimen*⟩ register `\fb@lblwidth` and then scans the bibliography and stores each reference in a global macro `\fb@r.`⟨*key*⟩ where ⟨*key*⟩ is the key of the reference. The references are read one token at a time so that we may detect `\verb` commands even if they are hidden in groups.

```
\newdimen\fb@lblwidth
```

```
\newcommand*\fb@thebibliography[1]{%
```

We open still a new group to prevent our definitions to conflict with macros that might be used by `\end{thebibliography}`. We let `\endthebibliography` equal to `\endgroup` so that it closes this group.

```
  \begingroup
  \let\endthebibliography\endgroup
```

Record the size of the longest label

```
  \settowidth\dimen@{\footrefstyle\footreflabel{#1}}%
  \ifdim\dimen@>\fb@lblwidth\global\fb@lblwidth=\dimen@\fi
```

`\@bracelevel` `\@bgrouplineno` `\@on@line` `\@eat`

Some definitions necessary to read the bibliography entries. All these definitions are local since the command `\begin{thebibliography}` opened a new group. They will be cancelled when `\end{thebibliography}` is executed. The names of global definitions have the form `\fb@...` while the names of local definitions simply start with `\@...` We reuse as much as possible existing global names so that we do not use memory unnecessarily. the `\@bracelevel` counter keeps track of groups nesting while reading the bibliography. The `\@bgrouplineno` counter holds the number of the input line where the current group started. It is used for error messages. `\@on@line` is similar to the `\on@line` command of the LaTeX 2_ε kernel but also shows the name of the current file.

```
  \newcount\@bracelevel
  \newcount\@bgrouplineno
  \def\@on@line{\on@line\space of \fb@bibname.bbl}%
  \def\@eat{\let\@tempa= }%
```

`\@bgroup` `\@egroup` `\bgroup` `\egroup`

`\bgroup` and `\egroup` are redefined so that we may distinguish between explicit and implicit begin or end group characters.

```
  \let\@bgroup{%
  \let\@egroup}%
  \let\bgroup\relax
  \let\egroup\relax
```

`\@actlet`

`\@actlet`\⟨*char*⟩⟨*cmd*⟩ makes ⟨*char*⟩ active and lets it equal to ⟨*cmd*⟩.

```
  \def\@actlet##1{%
      \catcode`##1=\active
      \begingroup\lccode`~=`##1\lowercase{\endgroup\let~}}%
```

`\@noitemerror`

`\@noitemerror` is called to raise an error if anything is seen between `\begin{thebibliography}` and the first `\bibitem`. If the user types ⟨*return*⟩ at the prompt, the next token is swallowed and the processing goes on.

```
  \def\@noitemerr{%
      \PackageError{footbib}{Missing \string\bibitem\@on@line}\@empty
      \afterassignment\@readbib\@eat}%
```

`\@errifbraces`

`\@errifbraces`⟨*cmp*⟩ compares `\@bracelevel` with 0 using ⟨*cmp*⟩ (= or >) and raises an error if the test succeeds.

```
\def\@errifbraces##1{%
    \ifnum\@bracelevel##1\z@
      {\ifx##1>\let\inputlineno\@bgrouplineno\fi
      \PackageError{footbib}{%
         \ifx##1>Unmatched begin\else Extra end\fi-group
         character\@on@line}\@empty}%
    \fi}%
```

Now come all the commands which read and handle the tokens.

`\@readbib` `\@xreadbib` These macros read the next token and take the appropriate action.

```
\def\@readbib{\futurelet\@tok\@xreadbib}%
\def\@xreadbib{%
    \ifx\@tok\@sptoken\let\@tempa\@readsp
    \else\ifx\@tok\par\let\@tempa\@readpar
    \else\ifx\@tok\bibitem\let\@tempa\@endbibitem
    \else\ifx\@tok\end\let\@tempa\@checkendbib
    \else\if@newlist\let\@tempa\@noitemerr
    \else\ifx\@tok\@bgroup\let\@tempa\@eat\afterassignment\@begingroup
    \else\ifx\@tok\@egroup\let\@tempa\@eat\afterassignment\@endgroup
    \else\ifx\@tok\verb\let\@tempa\relax
    \else\let\@tempa\@addtotxt
    \fi\fi\fi\fi\fi\fi\fi\fi\@tempa}%
```

`\bibitem` `\@bibitem` `\@lbibitem` The macro `\bibitem` is called if the next token is `\bibitem`. It calls `\@lbibitem` or `\@bibitem` depending on whether a label is provided or not. If no label is provided, `\@bibitem` provides one. The definition of this macro depends on the switch `\iffb@firstcite` which is true iff a `firstcite` option was used. If so, a *dynamic* label is provided. Otherwise, a *static* label is provided which is the current value of the counter `\fb@refcount`.

```
\def\bibitem{%
    \@errifbraces>%
    \@newlistfalse
    \@ifnextchar[\@lbibitem\@bibitem}%
\iffb@firstcite
  \def\@bibitem{\@lbibitem[\fb@setlbl]}%
\else
  \def\@bibitem{%
    \advance\fb@refcount\@ne
    \expandafter\@lbibitem\expandafter[\the\fb@refcount]}%
\fi
\def\@lbibitem[##1]##2{%
    \fb@lbl{{##1}}%
    \fb@vedef\fb@key{##2}%
    \fb@txt{}%
    \let\@lastsptok\@empty
    \@inlabeltrue
    \@readbib}%
```

`\@checkendbib` The macro `\@checkendbib` is called when the next token is `\end`. It reads the argument of `\end` and checks whether it is `thebibliography`. If so it calls `\@endbibitem` to terminate the current reference (if any) and reinserts `\end{thebibliography}` which will terminate the bibliography.

```
\def\@checkendbib\end##1{%
    \def\@tempa{##1}\def\@tempb{thebibliography}%
    \ifx\@tempa\@tempb
      \def\@tempa{\@endbibitem\end{##1}}%
    \else
      \if@newlist
        \let\@tempa\@noitemerr
      \else
        \def\@tempa{\@addtotxt{\end{##1}}}%
      \fi
    \fi\@tempa}%
```

`\@endbibitem` The macro `\@endbibitem` terminates the current reference (if any) and calls `\fb@setref` which stores it in a macro. If the switch `\iffb@xref` is true, i.e a `citeonce*` option was used, `\@endbibitem` adds to the text of the reference the command `\fb@settxt` which will modify it dynamically.

```
\def\@endbibitem{%
    \if@newlist\else
      \@errifbraces>%
      \iffb@xref
        \fb@txt\expandafter{\expandafter\fb@settxt\expandafter{\the\fb@txt}}%
      \fi
      \fb@setref
    \fi}%
```

`\@addtotxt` The macro `\@addtotxt` adds to the text of the reference so far, first the last *space* token (⟨*space*⟩ or `\par`) and then its argument.

```
\long\def\@addtotxt##1{%
    \fb@txt\expandafter\expandafter\expandafter
      {\expandafter\the\expandafter\fb@txt\@lastsptok##1}%
    \let\@lastsptok\@empty
    \@inlabelfalse
    \@readbib}%
```

`\@readsp` `\@readpar` `\@lastsptok` The *space* tokens ⟨*space*⟩ and `\par` are handled in a delayed way. They are first recorded in a single place (`\@lastsptok`), so that each one overrides the previous one. The most recent one is added to the text of the reference each time `\@addtotxt` is called. This mechanism discards any space preceding a `\par` token and also the `\par` which occurs generally at the end of each reference.

```
\expandafter\def\expandafter\@readsp\space{%
    \if@inlabel\else\let\@lastsptok\space\fi
    \@readbib}%
\def\@readpar\par{%
    \if@inlabel\else\def\@lastsptok{\par}\fi
    \@readbib}%
```

\@begingroup
\@endgroup

The macros \@begingroup and \@endgroup are called when a begin- or end-group character is seen. \@begingroup opens a new group and increments the counter \@bracelevel. It also records the number of the input line which may be used later for error reporting. \@endgroup closes the group, which restores the previous value of \@bracelevel, and calls \@addtotxt to add the group to the text of the reference so far.

```
\def\@begingroup{%
    \@bgroup
    \advance\@bracelevel\@ne
    \@bgrouplineno=\inputlineno
    \fb@txt{}%
    \let\@lastsptok\@empty
    \@readbib}%
\def\@endgroup{%
    \@errifbraces=%
    \edef\@tempa{\@egroup\noexpand\@addtotxt{{\the\fb@txt\@lastsptok}}}%
    \@tempa}%
```

\verb
\@sverb

We provide special support for the \verb command. The following code is essentially borrowed from the LaTeX 2_ε kernel. Just we let the active characters equal to \relax so that they are not expanded.

```
\def\verb{%
    \begingroup
```

First make all special characters 'other'

```
    \let\do\@makeother \dospecials
```

Avoid ligatures

```
    \def\do####1{\@actlet####1\relax}\verbatim@nolig@list
```

An end of line character in the argument of \verb is an error.

```
    \@actlet\^^M\verb@eol@error
    \@ifstar{\@sverb*}{\@actlet\ \relax\@sverb\@empty}}%
\def\@sverb##1##2{%
    \@actlet##2\verb@egroup
```

Read the argument of \verb.

```
    \edef\@tempa{\noexpand\verb##1\noexpand~\iffalse}\fi}%
```

\verb@egroup

Normal termination of \verb. The ~ must be protected against expansion because we are still inside the \edef started by \@sverb.

```
\def\verb@egroup{%
    \noexpand~\iffalse{\fi}%
    \expandafter\endgroup\expandafter\@addtotxt\expandafter{\@tempa}}%
```

\verb@eol@error

The macro \verb@eol@error is called if an end of line character occurs before the normal termination of \verb. Unlike it is done in the LaTeX 2_ε kernel, we do not terminate the \verb because the most frequent case is when BibTeX breaks the argument of \verb because the line is too long. The closing delimiter will generally be found on the next line.

```
\def\verb@eol@error{\iffalse{\fi}%
    \GenericError\@empty{LaTeX Error:
        \noexpand\verb ended by end of line\@on@line}\@gobble%
        {The argument of \string\verb\@on@line\space contains
             an end of line.\MessageBreak
         Type \space X <return> \space to quit.\MessageBreak}%
```

We try to recover from error in case the user types ⟨*return*⟩ at the prompt. We where defining `\@tempa` which contains now `\verb`?`~`... where ? stands for `*` or nothing.

```
    \toks@\expandafter{\@tempa}%
    \edef\@tempa{\the\toks@\iffalse}\fi}%
```

Some initialisation before starting to read the bibliography.

```
\@bracelevel\z@
\@newlisttrue
\@inlabeltrue
```

And finally start to read the bibliography. This is the end of `\fb@thebibliography`

```
\@readbib}
```

At the end of the document, we first do a `\clearpage` to be sure that all the writes have been performed. Then we close the auxiliary file and finally read it to check whether any change occured between the previous run and the current one. If so, we issue a warning.

```
\AtEndDocument{%
  \clearpage
  \if@filesw
    \immediate\closeout\fb@auxout
    \begingroup
      \let\citation\@gobble \let\bibstyle\@gobble \let\bibdata\@gobble
      \def\bibcite#1#2#3#4#5{%
        \advance\fb@citecount\@ne
        \fb@vedef\@tempa{#1}%
        \edef\@tempa{{\@tempa}{#2}{#3}{#4}{#5}}%
        \expandafter\ifx\csname fb@c.\the\fb@citecount\endcsname
          \@tempa\else\@tempswatrue\fi}%
      \fb@citecount\z@
      \@tempswafalse
      \input\fb@bibname.aux
      \if@tempswa
        \PackageWarning{footbib}{Bibliography not yet stable. Rerun
                                    LaTeX\@gobble}%
      \fi
    \endgroup
  \fi}
```

4.11 Output routine

We redefine the macros `\@specialoutput`, `\@doclearpage`, `\@makecol` and `\@reinserts` used by the output routine of LaTeX so that we may insert the foot bibliography. The code is just a patch to the macros defined in the LaTeX 2_ε kernel.

`\@specialoutput` If a foot bibliography is present, `\@specialoutput` must add to the height of the page the height plus depth of the foot bibliography and the length of the skip above it.

```
\def\@specialoutput{%
  \ifnum \outputpenalty>-\@Mii
    \@doclearpage
  \else
    \ifnum \outputpenalty<-\@Miii
      \ifnum \outputpenalty<-\@MM \deadcycles \z@ \fi
      \global \setbox\@holdpg \vbox {\unvbox\@cclv}%
    \else
      \global \setbox\@holdpg \vbox{%
                      \unvbox\@holdpg
                      \unvbox\@cclv
                      \setbox\@tempboxa \lastbox
                      \unskip}%
      \@pagedp \dp\@holdpg
      \@pageht \ht\@holdpg
      \unvbox \@holdpg
      \@next\@currbox\@currlist{%
        \ifnum \count\@currbox>\z@
          \advance \@pageht \@pagedp
          \ifvoid\footins \else
            \advance \@pageht \ht\footins
            \advance \@pageht \skip\footins
            \advance \@pageht \dp\footins
          \fi
```

footbib addition

```
          \ifvoid\fb@ins\else
            \advance\@pageht\ht\fb@ins
            \advance\@pageht\skip\fb@ins
            \advance\@pageht\dp\fb@ins
          \fi
```

```
          \ifvbox \@kludgeins
            \ifdim \wd\@kludgeins=\z@
              \advance \@pageht \ht\@kludgeins
            \fi
          \fi
          \@reinserts
          \@addtocurcol
        \else
```

```
          \@reinserts
          \@addmarginpar
        \fi}%
       \@latexbug
      \ifnum \outputpenalty<\z@
        \if@nobreak
          \nobreak
        \else
          \addpenalty \interlinepenalty
        \fi
      \fi
    \fi
  \fi}
```

`\@doclearpage` The test at the beginning of `\@doclearpage` has been modified so that it checks that *both* footnotes and the foot bibliography are empty.

```
\def \@doclearpage {%
```

`footbib` modification

```
%  \ifvoid\footins
  \@tempswatrue
  \ifvoid\footins\else\@tempswafalse\fi
  \ifvoid\fb@ins\else\@tempswafalse\fi
  \if@tempswa
```

```
    \ifvbox\@kludgeins
      {\setbox \@tempboxa \box \@kludgeins}%
    \fi
    \setbox\@tempboxa\vsplit\@cclv to\z@ \unvbox\@tempboxa
    \setbox\@tempboxa\box\@cclv
    \xdef\@deferlist{\@toplist\@botlist\@deferlist}%
    \global \let \@toplist \@empty
    \global \let \@botlist \@empty
    \global \@colroom \@colht
    \ifx \@currlist\@empty
    \else
      \@latexerr{Float(s) lost}\@ehb
      \global \let \@currlist \@empty
    \fi
    \@makefcolumn\@deferlist
    \@whilesw\if@fcolmade \fi{\@opcol\@makefcolumn\@deferlist}%
    \if@twocolumn
      \if@firstcolumn
        \xdef\@dbldeferlist{\@dbltoplist\@dbldeferlist}%
        \global \let \@dbltoplist \@empty
        \global \@colht \textheight
        \begingroup
          \@dblfloatplacement
          \@makefcolumn\@dbldeferlist
          \@whilesw\if@fcolmade \fi{\@outputpage\@makefcolumn\@dbldeferlist}%
```

```
644         \endgroup
645       \else
646         \vbox{}\clearpage
647       \fi
648     \fi
649   \else
650     \setbox\@cclv\vbox{\box\@cclv\vfil}%
651     \@makecol\@opcol
652     \clearpage
653   \fi}
```

`\@makecol` In addition to footnotes (if any), `\@makecol` must add the foot bibliography to the page. It is added immediately below the footnotes. The test at the beginning of `\@makecol` has been modified in the same way as in `\@doclearpage`

```
654 \def \@makecol {%
```

`footbib` modification

```
655 %  \ifvoid\footins
656   \@tempswatrue
657   \ifvoid\footins\else\@tempswafalse\fi
658   \ifvoid\fb@ins\else\@tempswafalse\fi
659   \if@tempswa
```

```
660     \setbox\@outputbox \box\@cclv
661   \else
662     \setbox\@outputbox \vbox {%
663       \boxmaxdepth \@maxdepth
664       \unvbox \@cclv
```

`footbib` addition

```
665       \ifvoid\footins \else
```

```
666         \vskip \skip\footins
667         \color@begingroup
668           \normalcolor
669           \footnoterule
670           \unvbox \footins
671         \color@endgroup
```

`footbib` addition

```
672       \fi
673       \ifvoid\fb@ins\else
674         \vskip\skip\fb@ins
675         \color@begingroup
676           \normalcolor
677           \footbibrule
678           \unvbox\fb@ins
679         \color@endgroup
680       \fi
```

```
      }%
  \fi
  \let\@elt\relax
  \xdef\@freelist{\@freelist\@midlist}%
  \global \let \@midlist \@empty
  \@combinefloats
  \ifvbox\@kludgeins
    \@makespecialcolbox
  \else
    \setbox\@outputbox \vbox to\@colht {%
      \@texttop
      \dimen@ \dp\@outputbox
      \unvbox \@outputbox
      \vskip -\dimen@
      \@textbottom}%
  \fi
  \global \maxdepth \@maxdepth}
```

`\@reinserts` The macro `\@reinsert` was modified to reinsert also the foot bibliography after float processing.

```
\def\@reinserts{%
  \ifvoid\footins\else\insert\footins{\unvbox\footins}\fi
```

footbib addition

```
  \ifvoid\fb@ins\else\insert\fb@ins{\unvbox\fb@ins}\fi
  \ifvbox\@kludgeins\insert\@kludgeins{\unvbox\@kludgeins}\fi}

\endinput
```

⟨/package⟩