footmisc —

a portmanteau package

for customising footnotes in \LaTeX\*

Robin Fairbairns†

June 6, 2011

Copyright statement

Program: footmisc.dtx

This program is offered under the terms of the LaTeX Project Public License, version 1.3c of this license or (at your option) any later version. The latest version of this license is in http://www.latex-project.org/lppl.txt, and version 1.3c or later is part of all distributions of LaTeX version 2005/12/01 or later.

This work has the LPPL maintenance status ‘author-maintained’.

History

This package originated as support of a personal project, which I was switching to \LaTeX\ 2e over the Christmas holiday period of 1993, using the first \beta release.

In its first form, it was known as the “footnote” package, but by the time I had released it to CTAN, that name had already been used by a package written by Mark Wooding. So the package is now known (as you can see) as “footmisc”.

1 User interface — package options

The footmisc package provides several different customisations of the way footnotes are represented in \LaTeX\ 2e documents (the sources of the code in this package are various, but all of it has been massaged by the author; where the code comes from elsewhere, there are attributions given below, somewhere or other).

The interface to the package’s options is mostly rather simple — each one is presented as an option in the \usepackage command, and for most, nothing else needs to be done. For example, to use a useful and consistent set, the author invokes the package with the command \usepackage[perpage,para,symbol*]{footmisc}.

For a small number of options, there are additional parameters available; these are described in the subsections below.

*This file has version number v5.5b, last revised 2011/06/06
†University of Cambridge Computer Laboratory, William Gates Building, J. J. Thompson Avenue, Cambridge, CB2 0FD, UK (rf10<at>cam.ac.uk)
1.1 Option perpage

This option resets footnote numbering for each page of the document. It needs at least two passes to do this correctly (though it comes as close as possible on the first pass). You generally have to make two passes with \TeX anyway, to get the cross-references right, so an additional pass for this purpose shouldn’t cause any additional problem. The option includes code to report that ‘Label(s) may have changed’, which will help the poor user to realise that (yet) another run is in order.

1.2 Option para

This option (derived from code by Dominik Wujastyk and Chris Rowley) causes footnotes to be typeset as a single paragraph at the bottom of the page on which they occur. In the case that there is only one footnote on the page, no effect will be observed. However, if there are several footnotes on the page, they will be run together in the page foot, each introduced by its footnote mark. The original demand for the option came from the needs of those preparing critical editions; such documents typically have large numbers of small footnotes, which look ridiculous if each is typeset in a paragraph of its own; in most other disciplines, such multiplicities of footnotes represent mere self-indulgence: the author of this package is disgracefully guilty of this.

Please note that “old” \TeX installations may have problems with the algorithm for para footnotes on very wide pages (for example, those used by the a0poster class). Recent \TeX installations use an improved technique that is believed not to be susceptible to this problem.

1.3 Option side

This option (suggested by Frank Mittelbach) causes footnotes to be typeset using the \marginpar command: this has the advantage that the note appears close to its “call-up”, but has all the disadvantages associated with the \marginpar command (which consumes ‘float’ slots, and doesn’t always place itself correctly at the top of pages in two-sided documents). Since the measure in which the footnote is to be typeset is likely to be pretty narrow, users of the side option are recommended also to use the ragged option, to avoid ugly spacing and line breaks.

There is a further problem (apart from the occasional failure to place the marginal note on the correct side of the page) in two-sided documents: one would like ‘raggedness’ to appear differently in different margins (setting the left, rather than the right, side ragged in the left margin). (The author would welcome suggestions on means of addressing the problem.)

1.4 Option ragged and \footnotelayout

The package provides facilities for ragged right setting of footnotes (so long as the para option isn’t in effect). The change is effected by use of the command \footnotelayout; the package inserts this command into the start of the argument of \footnotetext (in effect: \footnote works, roughly, by calling the guts of \footnotetext at its end).
If you want to use some special effect other than ragged right, feel free to change \footnotelayout yourself: some intriguing (and completely undesirable) results are no doubt available. Change the setting simply by use of \renewcommand\footnotelayout... The \ragged option simply sets \footnotelayout to set \raggedright or \RaggedRight as appropriate. (If you intend to use the \ragged2e package, load it before footmisc — if footmisc finds \RaggedRight is available, it automatically uses it in place of \raggedright.)

1.5 Option symbol

This option simply establishes that footnotes are ‘labelled’ by a symbol sequence. The command used is equivalent to that suggested in \LaTeX manuals such as Lamport’s (the job performed by the option is very simple, and doesn’t really need a package).

Using symbols to ‘number’ your footnotes can be problematic: there is a limited number of symbols, and \LaTeX will report an error if your footnotes exceed that limit. To avoid such problems, consider the symbol* option, or the \setfnsymbol command (see the next two sections), or number your footnotes by the page (see section 1.1).

1.6 Option symbol*

This is the symbol option, but with protection against the tedium that arises because of the instability of the perpage option. When executing the perpage option, the package often allocates footnotes to the wrong pages, only to correct itself on a later run (having warned the user of the need for the later run with a ‘Label(s) may have changed’ message). In these circumstances the symbol option is prone to producing \LaTeX errors, which stop processing, and confound automatic generation procedures. In the same situation, the symbol* option produces information messages and a warning message at end document, and the user may scan the log for those messages after processing has stabilised. The option produces numbers (17 and higher, in the case of the default symbol set) in place of symbols, when the footnote number is too large.

1.7 The \setfnsymbol and \DefineFNsymbols commands

These commands permit the definition and use of alternative (ordered) sets of symbols for numbering footnotes. \LaTeX of course comes with such a set ready-defined, but the choice of symbols isn’t universally loved. You may define a set of symbols with the \DefineFNsymbols command. \LaTeX’s default set would be defined by the command:

\DefineFNsymbols*{lamport}{\dagger\dagger}{\ddagger\ddagger}{\P\|}{\S\%}{\dagger\dagger}\{\dagger\dagger\ddagger\ddagger\}

Defined this way, the symbol set produces a “counter too large” error; a robust version of the set (cf. the symbol* option (see 1.6) using the \DefineFNsymbols command without the optional *). You may select a set of symbols by use of the \setfnsymbol command; so to restore use of the default set, you would type:

\setfnsymbol{lamport}
This package defines a small selection of alternative sets of symbols, using \DefineFNsymbol:

\begin{verbatim}
bringhurst  * † ‡ § || ¶
chicago     * † ‡ § || #
wiley       * ** † ‡ § ¶ ||
\end{verbatim}

Together with a version of Lamport’s original set that, with doubled versions of § and ¶, and tripled versions of everything but the vertical bars, provides a symbol range to cover counters up to 16.

This last set, known as lamport* is selected as the default symbol set by the package.

1.8 Option bottom

This option forces footnotes to the bottom of the page; this is only noticeably useful in case that \raggedbottom is in effect, when L\TeX would normally set the footnotes a mere \skip\footins distant from the bottom of the text.

There’s a further infelicity in L\TeX’s placing of footnotes of the bottom of pages: if a bottom float appears on a page, L\TeX places the footnote above it. The bottom option places the footnote at the foot of the page.

1.9 Option marginal

This option adjusts the position of footnote mark relative to the start of the line in which they appear (the the option is incompatible with option para, for obvious reasons).

When this option is in effect, the footnote is set \footnotemargin relative to the left margin of the page; the default setting for \footnotemargin is -0.8em, which means that the footnote mark will be set jutting 0.8em into the margin. If \footnotemargin is a positive length, the footnote mark will be set with its right edge \footnotemargin from the margin. (In the absence of the option, \footnotemargin is set to 1.8em; you may change that value with a \setlength command.)

1.10 Option flushmargin

This option is as option marginal, but sets the footnote marker flush with, but just inside the margin from, the text of the footnote.

1.11 Option hang

This option sets the footnote mark flush with the margin, and makes the body of the footnote hang at an indentation of \footnotemargin (if that is a positive distance), or the width of the marker (if \footnotemargin \leq 0). The option code itself leaves \footnotemargin at its default value of 1.8em.

The footnote itself may of course be longer than one paragraph; if so, the paragraphs will be separated by the vertical space specified by \hangfootparskip, and the second and subsequent paragraphs are indented by \hangfootparindent. Default values are:
The user may redefine these values (using \renewcommand): it is best to use the font-size-dependent measures (multiples of \baselineskip for the skip, multiples of \em for the indent). Note that the default has only one of the two values non-zero; both zero may result in easily-missed paragraph breaks, and both non-zero is not generally thought to be a good-looking option.

1.12 Option norule
This option suppresses the ‘normal’ footnote rule, and advances \skip\footins a bit to compensate

1.13 Option splitrule
This option makes puts a full-width rule above the split-off part of a split footnote. (Remember that split footnotes don’t happen if you’re doing paragraph footnotes.)

The option provides three different \footnoterule commands:
- \mpfootnoterule for use in minipages
- \pagefootnoterule for normal footnotes on regular pages
- \splitfootnoterule for the tail of a split footnote

By default, \mpfootnoterule and \pagefootnoterule retain the original definition of \footnoterule (which may have been modified by a norule option), while \splitfootnoterule becomes a full-width rule.

1.14 The stable option
This option deals with the problem of placing footnotes in section titles (and so on). While there is (sometimes, just) justification for putting footnotes in titles, \TeX’s treatment of the content of titles militates against them. Of course, the title argument is ordinarily a moving one, and \footnote is a fragile command, but the real problem comes from the way the argument actually moves — which is to two places. The argument moves to the table of contents, where the footnote will (at least) look odd. But the argument also moves to the marks that make up page headers, etc., and there it creates havoc, since page headers are executed in page make-up, and page make-up must not create footnotes.

If you use the stable option, the footnote won’t move to the table of contents or the page headers, but it will be typeset correctly within the title itself.

The situation with \footnotemark is less dire (it could in principle appear in page headers, for example); footnote marks appearing on pages other than where their text appears are none the less confusing, and the stable option treats \footnotemark in the same way that it treats \footnote.

1.15 The multiple option
This option deals with the case where the author needs to type things like

\mumble\footnote{blah}\footnote{grumble}
Without special treatment, \LaTeX{} would output something like
\texttt{mumble}\textsuperscript{13,14}

What the \texttt{multiple} option makes of the above is
\texttt{mumble}\textsuperscript{13,14}

which is what most people would expect. The comma separator actually derives
from the definition of \texttt{\multfootsep}, which may be changed by \texttt{\renewcommand}
if the option is in effect.

The option also treats \texttt{\footnotemark} in the same way.

\section*{1.16 User interface — miscellaneous commands}

The package also defines some miscellaneous footnote-related commands. The
present group provides alternative means of producing footnote marks: \texttt{\footref}
and \texttt{\mpfootnotemark}.

When you're in a minipage, \texttt{\footnote} numbers run according to the minipage's own footnote counter, and the marks are set in italic letters. However, the
numbers used by \texttt{\footnotemark} make reference to the 'main' footnote counter,
and are set in whatever is the current style for that: this behaviour often surprises,
and there's no obvious way in standard \LaTeX{} to “get around” it. The command
\texttt{\mpfootnotemark} gets around this problem in a minipage, by generating footnote
marks in the same way as those used by \texttt{\footnote}.

In fact, making reference to footnotes in general can be problematic: it can be
done by noting down the value of the footnote marker in a counter (or the like)
and then using the value in a subsequent \texttt{\footnotemark} or \texttt{\mpfootnotemark}.
This is a tedious way of going about things, and doesn’t allow representation of
all possible forms of footnote mark; \texttt{\footref} is a form of reference command
that sets the reference as if it were a footnote. The label should be set \textit{within} the
argument of the footnote command that is being labelled:

\begin{verbatim}
...\footnote{Note text\label{fnlabel}}
...
... potato head\footref{fnlabel}
\end{verbatim}

\section*{2 User interface — interactions with other packages}

The \texttt{footmisc} package modifies several parts of the \LaTeX{} kernel; what gets modified
depends on the options you select. This behaviour can cause problems with other
packages, particularly those that also modify the kernel.

Known interactions are:

\texttt{setspace} The \texttt{setspace} package modifies the way line spacing is calculated in foot-
notes. \texttt{Footmisc} knows about this, and preserves the change. However, you
\textit{must} load \texttt{setspace} \textit{before} \texttt{footmisc}.

\texttt{memoir} class The class emulates \texttt{setspace}, and we detect that emulation and deal
with it in the same way as \texttt{setspace}.
The `hyperref` package has ambitions to make hyperlinks from footnote marks to the corresponding footnote body; naturally this causes grief to `footmisc`, and unfortunately no remedy is currently known. If you use `footmisc`, suppress `hyperref`'s hyper-footnotes, by loading it as:
\usepackage[hyperfootnotes=false,...]{hyperref}

Further work on the interaction between the two packages is proposed, but not yet scheduled.

The `manyfoot` package permits several independent sequences of footnotes. Some preliminary work towards interworking with `footmisc` has been completed, but more remains to be done at the time of writing.

### 3 Code: Preliminaries

Well — here we go: let’s make the package file:

1 \(\langle \ast package \rangle\)

Now declare what environment we need:

2 \NeedsTeXFormat{LaTeX2e}[1994/12/01]

We need a token register in case we have to patch \@makecol:

3 \newtoks\FN@temptoken

\protected@writeaux

This command is defined for future compatibility with Matt Swift’s `newclude` package (still, after all this time, not out of beta status).

4 \providecommand\protected@writeaux{%
5 \protected@write\@auxout
6 }

\l@advance@macro

This command is defined for future compatibility with Matt Swift’s `newclude` package (still, after all this time, not out of beta status).

7 \def\l@advance@macro{\@advance@macro{\edef}]
8 \def\l@advance@macro\#1\#2\#3{\expandafter\@tempcnta\#2\relax
9 \@advance@tempcnta\#3\relax
10 \#1\#2\{\the\tempcnta\%
11 }

Now we define a jolly little macro to advance a macro count (#1) by a given amount (#2).

12 \let\l@advance@macro\l@advance@macro

\iffFN@etex

Check whether we’re using etex

13 \newif\iffFN@etex
14 \ifx\dimexpr\undefined
15 \FN@etexfalse
16 \else
17 \FN@etextrue
18 \fi

\footnotemargin

Finally, we define the length used by the `marginal` option, and initialise it as if we’ve not had the option.

19 \newdimen\footnotemargin
20 \footnotemargin1.8em\relax
4 Package options

Most of the code of the package is contained within the option processing, one way or another (that which isn’t, is executed after \ProcessOptions as a result of flags set in the option processing).

4.1 The symbol option

This is a declaration that appears in the original \TeX{} book. Since it appeared in the old pagefoots.sty (presumably since it goes so naturally with the perpage option), I've added this trivial piece of customisation to the package.

\begin{verbatim}
\DeclareOption{symbol}{\renewcommand\thefootnote{\fnsymbol{footnote}}}
\end{verbatim}

4.2 The symbol* option

The robust version of the symbol option: if the current ‘symbol’ option doesn’t provide enough variants, use arabic footnote number. We use a robust version of the “extended ordinary” symbol set, described later (in section 1.7).

\begin{verbatim}
\DeclareOption{symbol*}{% 
\renewcommand\thefootnote{\@fnsymbol\c@footnote}\
\AtEndOfPackage{\setfnsymbol{lamport*-robust}}%
}
\end{verbatim}

4.3 The para option

The basis of the code for this option comes from \TeX{}book, p.398 ff. (“Dirty Tricks”), though it does (of course) avoid redefining \ which has some other (somewhat significant) uses in \TeX{}! The user should be aware of Knuth’s note on the limitations of this method of doing the job: the \TeX{} stack is used four times per footnote, and the stack is limited (see the \TeX{}book, p.300 ff.). If you have very large numbers of footnotes (in the hundreds), and encounter the error “! \TeX{} capacity exceeded, sorry (... save size ...)", you may need to break your text into smaller sections and compile the separately. Fortunately (say the comments on the original fnpara.sty) this is very easy to do with \TeX{}, provided that you reset the footnote counter to make the joins seamless.

\begin{verbatim}
\ifFN@para Define the para option: now simply sets a marker for use later when defining the option’s auxiliary code and when patching the output routine and so on.
\newif\ifFN@para \FN@parafalse
\DeclareOption{para}{\ifFN@sidefn
\PackageError{footmisc}{Option "\CurrentOption" incompatible with \option{"side"}}%
\else
{I shall ignore "\CurrentOption"}\
\fi
\FN@paratrue

\end{verbatim}
4.4 The \texttt{side} option

\ifFN@sidefn
Simply changes the behaviour of \texttt{@footnotetext}; incompatible with paragraph footnotes.
\newif\ifFN@sidefn \FN@sidefnfalse
\DeclareOption{side}{\ifFN@para
\PackageError{footmisc}{Option "\CurrentOption" incompatible with option "para"}%
{I shall ignore "\CurrentOption"}%
\else
\FN@sidefntrue
\fi}

4.5 The \texttt{ragged} option

\texttt{\footnotelayout}
A very simple option that merely changes the definition of one macro. Note detection of the presence of the \texttt{ragged2e} package.
\let\footnotelayout\@empty
\DeclareOption{ragged}{\ifundefined{RaggedRight}%
{\renewcommand\footnotelayout{\linepenalty50 \raggedright}}%
{\renewcommand\footnotelayout{\linepenalty50 \RaggedRight}}%
}

4.6 The \texttt{perpage} option

\ifFN@perpage
A footnote-numbering modification: a new algorithm replacing one from Brian T. Schellenberger, which has proved to be flawed. We simply set a marker here, and define code later depending on the state of the marker (see section 5.4).
\newif\ifFN@perpage \FN@perpagefalse
\DeclareOption{perpage}{\FN@perpagetrue}

4.7 The \texttt{PPdebug} option

\ifFN@pp@debug
Sets a flag; the messages are generated in various places throughout the code. The option is not available in the package as distributed: modify the \texttt{.ins} file to generate a version of the package that includes the option, if you feel you need it.
\newif\ifFN@pp@debug \FN@pp@debugfalse
\DeclareOption{PPdebug}{\FN@pp@debugtrue}

4.8 The \texttt{bottom} option

\ifFN@bottom
All this needs to do is to set a flag to say that it should happen
\newif\ifFN@bottom \FN@bottomfalse
\DeclareOption{bottom}{\FN@bottomtrue}
4.9 The marginal option
Again, the processing of the option is pretty trivial:
\DeclareOption{marginal}{%
  \footnotemargin-0.8em\relax
%
}

4.10 The flushmargin option
Again, the processing of the option is pretty trivial:
\DeclareOption{flushmargin}{%
  \footnotemargin0pt\relax
%
}

4.11 The hang option
\ifFN@hangfoot
  We need a switch, since \@makefntext needs to be patched.
\newif\ifFN@hangfoot \FN@hangfootfalse
\DeclareOption{hang}{%
  \FN@hangfoottrue
%
}\hangfootparskip \hangfootparindent
Layout parameters for hanging footnotes; \hangfootparskip and \hangfootparindent are (respectively) values to use for \parskip and \parindent when in hanging footnotes.
\newcommand*{\hangfootparskip}{0.5\baselineskip}
\newcommand*{\hangfootparindent}{0em}%

4.12 The norule option
Pretty simple too...
\DeclareOption{norule}{%
\renewcommand*{\footnoterule}{}%
\advance\skip\footins 4\p@@plus2\p@\relax
%
}

4.13 The splitrule option
\split@prev
This is from a posting by Donald Arseneau dated 13 November 1996. The code relies on the fact that \LaTeX only uses inserts for footnotes, so that if any insert is going to be split, it’s going to be a footnote.
\DeclareOption{splitrule}{%
\gdef\split@prev{0}
%
}\pagefootnoterule \mpfootnoterule \splitfootnoterule
Define defaults for the three footnote rules: note, we inherit the current state of \footnoterule for the two ‘regular’ footnote defaults, and if we’ve been preceded by option norule, they will both become null...
\let\pagefootnoterule=\footnoterule
\let\mpfootnoterule=\footnoterule
\def\splitfootnoterule{\kern-3\p@ \hrule \kern2.6\p@}
Now redefine \footnoterule to distinguish the three situations.

\def\footnoterule{\relax
  \ifx \@listdepth\@mplistdepth
    \mpfootnoterule
  \else
    \ifnum\split@prev=\z@ \pagefootnoterule \else \splitfootnoterule \fi
  \fi
}\fi

\ifFN@stablefootnote
  \newif\ifFN@stablefootnote \FN@stablefootnotefalse
  \DeclareOption{stable}{\FN@stablefootnotetrue}
\fi

\ifFN@multiplefootnote
  \newif\ifFN@multiplefootnote \FN@multiplefootnotefalse
  \DeclareOption{multiple}{\FN@multiplefootnotetrue}
\fi

\ProcessOptions

\section{Hacking kernel commands}

Various standard commands (some of them internal ones) need to be hacked to achieve our effects, and we do all of this now, according to flags set in option processing.
5.1 The output routine

Now; do we need to mess about with the output routine? If either \texttt{para} or \texttt{bottom} has been invoked, we do.

\begin{verbatim}
\let \if@tempswa \ifFN@bottom
\iffFN@para \@tempswa \true \fi
\fi
... so we've patching to do.

First, we ensure that \texttt{@makecol} is as expected from the time at which these macros were written: since we're going to patch it, we had better be sure that we're patching the right thing. (There was a minuscule change to the definition 1999, but this doesn't as far as I can tell make any difference to the semantics of the definition we base our patch on.)

\begin{verbatim}
\if@tempswa\fmtversion{2005/12/01}%
\CheckCommand*{\@makecol}{\ifvoid \footins
\setbox\@outputbox \box\@cclv
\else
\setbox\@outputbox \vbox{\setboxmaxdepth \@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}
\global\maxdepth\@maxdepth
\fi
\fi
\else\fmtversion{2003/12/01}%
\CheckCommand*{\@makecol}{\ifvoid \footins
\setbox\@outputbox \box\@cclv
\else
\setbox\@outputbox \vbox{\boxmaxdepth \@maxdepth
\@tempdima \dp\@cclv
\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}
\global\maxdepth\@maxdepth
\fi
\fi
\end{verbatim}
\end{verbatim}

12
If we’re doing paragraph footnotes, the output routine needs different code to place the actual text. We prepare this code here, since it’s potentially used in two different places.

We prepare the code in a token register to be used at the appropriate place in the patching of \@makecol; thus it becomes a token register containing code to place stuff in a token register.

\ifFN@para

We make a box out of the paragraph of footnotes, and then stuff the contents of the box into that which is going to be \shipped out.

\FN@temptoken
\toks@\expandafter{\the\toks@}
\vskip\skip\footins
\color@begingroup
\normalcolor\footnoterule
\global\setbox\FN@tempboxc\vbox{\makefootnoteparagraph}
\unvbox\FN@tempboxc
\color@endgroup

\fi
\global\maxdepth@maxdepth
}\global\maxdepth\@maxdepth
}

If we’re not doing paragraph footnotes, we insert the little bit of code that would have been replaced by the stuff above:

\else
Now we start building up the revised version of \@makecol. The definition starts out in \toks@; first the bottom version:
\ifFN@bottom
\toks@\setbox\@outputbox \box\@cclv
\xdef\@freelist{\@freelist\@midlist}\
global\let\@midlist\@empty
@combinefloats
\ifvoid\footins
\setbox\@outputbox \vbox\bgroup
\boxmaxdepth\@maxdepth
\unvbox\@cclv
\vfill\relax
\the\FN@temptoken
\toks@\expandafter{\the\toks@\egroup}\fi
\fi
\global\let\@midlist\@empty
@combinefloats
\fi

Not putting stuff at the bottom: footnotes are placed using the kernel’s algorithm.
\ifvoid\footins
\setbox\@outputbox \vbox\bgroup
\boxmaxdepth\@maxdepth
\unvbox\@cclv
\vfill\relax
\the\FN@temptoken
\toks@\expandafter{\the\toks@\egroup}\fi
\else
\setbox\@outputbox \vbox\bgroup
\boxmaxdepth\@maxdepth
\unvbox\@cclv
\vfill\relax
\the\FN@temptoken
\toks@\expandafter{\the\toks@\egroup}\fi

Finally, close the \setbox and the \ifvoid and tag the parts of the definition of \@makecol up to the end of the definition of the bottom version on to \toks@.
\toks@\expandafter{\the\toks@\egroup}\fi
\xdef\@freelist{\@freelist\@midlist}\
global\let\@midlist\@empty
@combinefloats
}\fi

Finally, create the new definition from the resulting object with the remainder of the original \@makecol tagged on at the end.
\toks@\expandafter{\the\toks@\ifvbox\@kludgeins
All of the above occurred conditionally on the 'or' of \ifFN@para and \ifFN@bottom, so we now close the conditional.
\fi

5.2 The requirements of \@footnotetext

\ifFN@baselinestretch
\FN@singlespace
Whatever we do, we are going to patch \@footnotetext; so first of all, we'll check it's not been hacked by anyone other than setspace.sty (while we're at it we also record whether setspace is loaded). so we do this here:
\newif\ifFN@setspace
\@ifpackageloaded{setspace}{% 
\FN@setspacetrue
\@ifclassloaded{memoir}{% we're seeing memoir's emulation of setspace
\let\FN@baselinestretch\m@m@singlespace
}% we're seeing setspace in its own right
\let\FN@baselinestretch\setspace@singlespace
}%
\FN@setspacelfalse
%
There's substantial patching to be done if we're doing paragraph footnotes:
\ifFN@para
\renewcommand\@footnotetext[1]{% 
insert compatibility code with setspace.sty if necessary
\ifFN@setspace
\let\baselinestretch\FN@baselinestretch
\fi
\reset@font\footnotesize
\interlinepenalty\interfootnotelinepenalty
\splittopskip\footnotesep
\splitmaxdepth \dp\strutbox
\floatingpenalty\@MM
\hsize\columnwidth
\@parboxrestore
\protect@edef\@currentlabel{\csname p@footnote\endcsname\@thefnmark}%  
\color@begingroup
We set the paragraph in an \hbox and apply the fudge factor here:

\setbox\FN@tempboxa=\hbox{%

This needs a parameter; the rule should be moved to the beginning of the footnote paragraph, but the \ignorespaces should be left here.

\makefntext{\ignorespaces#1\strut}

We insert a penalty here to help line breaking in the footnote paragraph; the value is taken from the \TeXbook.

\penalty-10\relax
\hskip\footglue
}\hbox{\ignorespaces#1\strut}

\iftex
\ht\FN@tempboxa=\dimexpr\wd\FN@tempboxa * \footnotebaselineskip / \columnwidth\relax
\else
\ht\FN@tempboxa=\fudgefactor\wd\FN@tempboxa
\fi
\box\FN@tempboxa
\color@endgroup
}

If we’re not doing paragraph footnotes, we now simply tag a \FN@mf@prepare command on the end of the definition; of course, there are different definitions according as whether we’re using side footnotes...

\else
\ifFN@sidefn
\renewcommand\@footnotetext[1]{%
\marginpar{insert compatibility code with \setspace.sty if necessary

\ifsprotect
\let\baselinestretch\FN@baselinestretch
\fi
\reset@font\footnotesize
\protected@edef\@currentlabel{\csname p@footnote\endcsname\@thefnmark
\color@begingroup
\makefntext{%
\ignorespaces#1\strut
\color@endgroup
}
\FN@mf@prepare
}
\else
\renewcommand\@footnotetext[1]{%
\insert\footins{17

17
insert compatibility code with \setspace if necessary
362 \ifFN@setspace
363 \let\baselinestretch\FN@baselinestretch
364 \fi
365 \reset@font\footnotesize
366 \interlinepenalty\interfootnotelinepenalty
367 \splitoppenalty\footnotepenalty
368 \splitmaxdepth \dp\strutbox
369 \floatingpenalty\@MM
370 \hsize\columnwidth
371 \@parboxrestore
372 \protected@edef\@currentlabel{%
373 \csname p@footnote\endcsname\@thefnmark
374 }%
375 \color@begingroup
376 \makefntext(\
377 \rule\z@\footnotepenalty
378 \ignorespaces#1\@finalstrut\strutbox
379 )%
380 \color@endgroup
381 )%
382 \FN@mf@prepare
383 )%
384 \fi
385 \fi

5.3 Support code for paragraph footnotes
This code used (most inefficiently) to be in the argument of the \DeclareOption;
this no doubt comes of that code having been written over Christmas 1993...
Now all executed under the \para conditional set in the option declaration.
386 \ifFN@para
\FN@tempboxa We need some temporary boxes, and \LaTeX only defines one
\FN@tempboxb 387 \let\FN@tempboxa\@tempboxa
\FN@tempboxb 388 \newbox\FN@tempboxa
\FN@tempboxb 389 \newbox\FN@tempboxb
\footglue A direct crib from the \TeXbook:
390 \newskip\footglue \footglue=1em plus.3em minus.3em
\@makefntext The standard classes set the footnote mark flush with the text of the footnote,
but that's not appropriate for paragraph footnotes, we find.
There's not much point in patching this code from the original, since the only
things it has in common with the original are the footnote mark and the footnote
text (which last is the argument). Note that the \leavevmode isn't necessary
except in the case of footnotes in minipages, which otherwise end up with the
\@makefntext being executed in restricted vertical mode, which results in its
\hbox ending up in a line of its own.
391 \long\def\@makefntext#1(\leavevmode
392 \@makefntext\nobreak
393 \hskip.5em\relax#1%
394 )
\footnotebaselineskip  We need to record a value for the baseline skip when in footnotes:

\newdimen\footnotebaselineskip
\footnotesize
\global\footnotebaselineskip=\normalbaselineskip

\fudgefactor  Now we derive a fudge factor from the baselineskip we’ve just established (we use \dimexpr if we’re in etex, so there’s no need for the fudge factor in that case).

\ifFN@etex
\else
\@tempdima=\footnotebaselineskip \multiply\@tempdima by 1024
\divide\@tempdima by \columnwidth \multiply\@tempdima by 64
\xdef\fudgefactor{\strip@pt\@tempdima }%
\fi

\makefootnoteparagraph  For use in the output routine

\long\def\makefootnoteparagraph{\unvbox\footins \makehboxofhboxes
\setbox\FN@tempboxa=\hbox{\unhbox\FN@tempboxa \removehboxes}
Now we are ready to set the paragraph:
\hsize\columnwidth
\@parboxrestore
\baselineskip=\footnotebaselineskip

\noindent
\rule{\z@}{\footnotesep}%
\unhbox\FN@tempboxa\par

\makehboxofhboxes \removehboxes  Support code for \makefootnoteparagraph

\def\makehboxofhboxes{\setbox\FN@tempboxa=\hbox{}}%
\loop
\setbox\FN@tempboxb=\lastbox
\if hbox\FN@tempboxb
\setbox\FN@tempboxa=\hbox{\box\FN@tempboxb \unhbox\FN@tempboxa}%
\repeat
\def\removehboxes{\setbox\FN@tempboxa=\lastbox
\if hbox
\FN@tempboxa\removehboxes%
\unhbox\FN@tempboxa
\fi
\fi

5.4  The other footnote commands

\ifFN@pp@footnotehint  A conditional needed by the perpage code: must be defined outside the perpage conditional
\newif\ifFN@pp@footnotehint
Counter used to store information about the next reset of the footnote number, in perpage mode.
\newcounter{pp@next@reset} \%

A conditional that mediates the interaction between the \texttt{perpage} option and the \texttt{multiple} option.
\newif\ifFN@pp@towrite\FN@pp@towritefalse\%

We prevent endless processions of diagnostics ‘footnote sequence lost’ by using this conditional (again, needs to be defined outside the \texttt{perpage} conditional:
\ifFN@pp@lastseq\FN@pp@writetemp\FN@pp@towritefalse\fi\%

Now, do we need to patch \texttt{footnote} for per-page footnotes?
\ifFN@perpage\CheckCommand*\texttt{footnote}\texttt{\ifnextchar [}%
  \% \stepcounter\@mpfn \protected@xdef\@thefnmark{\thempfn} %
  \@footnotemark \@footnotetext\%
}\renewcommand*\texttt{footnote}\texttt{\ifnextchar [}%
  \% \stepcounter\@mpfn \protected@xdef\@thefnmark{\thefootnote} %
  \FN@pp@footnote\@footnotemark %
In case that we’re \textit{not} running \texttt{multiple} option, \texttt{@footnotemark} won’t have written details to the .aux file, so do it now:
\ifFN@pp@towrite\FN@pp@writetemp\FN@pp@towritefalse\fi\%

And the analagous change for \texttt{footnotemark}
\CheckCommand*\texttt{footnotemark}%
  \% \stepcounter{footnote} %
  \protected@xdef\@thefnmark{\thefootnote} %
  \@footnotemark %
\renewcommand*\texttt{footnotemark}%
  \%
\FN@pp@initialstab

Now the supporting commands...

if we encounter no information in the .aux file, we make a first stab resetting footnote on the page number counter
\gdef\FN@pp@initial@stab{\@addtoreset{footnote}{page}}
\AtBeginDocument{\FN@pp@initial@stab}

We use a counter to keep pace with the footnotes: this counter is used in the data that's written to the .aux file, and matched to create the correct footnote numbers on the second and subsequent passes.
\newcounter{@fnserial}
\FN@pp@cpage

The package requires a “knowledge” of the current page number. It’s kept in \FN@pp@cpage
\def\FN@pp@cpage{0}
\footnotehint

However, the progress of page numbers isn’t predictable, so we have a flag saying ‘reset footnote number’. The flag is for indirect use by people who diddle with the page number, via the \footnotehint command, as well as various places where we know there could be a discontinuity.
\FN@pp@footnotehinttrue
\newcommand{\footnotehint}{%\setcounter{footnote}{0}%\protected@writeaux\relax{\protect{\FN@pp@footnotehinttrue}%.\tempcnta\c@\fnserial}%\advance\tempcnta\@ne%\global\c@pp@next@reset\tempcnta} %
\AtBeginDocument{\protected@writeaux\relax{\protect{\FN@pp@footnotehinttrue}%.\tempcnta\c@\fnserial}%\advance\tempcnta\@ne%\global\c@pp@next@reset\tempcnta} %
\FN@pp@footnotehintfalse

\FN@pp@lastfoot

Dummy value for the number of the last footnote we came across.
\def{\FN@pp@lastfoot}{-1}

\FN@pp@footnote@aux

The command \FN@pp@footnote@aux is written to the .aux file for every footnote counter allocated (other than in minipages):
#1 is the footnote serial number
#2 is the page the footnote was actually written on
\newcommand{\FN@pp@footnote@aux}[2]{%
\ifnum\FN@pp@lastfoot<1
  \if\FN@pp@footnotehint
    \FN@pp@resetfn{1}{#2}\%
    \FN@pp@footnotehintfalse
  \else
    \gdef\@tempa{#2}\%
    \ifx\@tempa\FN@pp@cpage
      \else
        \FN@pp@resetfn{1}{#2}\%
    \fi
  \fi
  \def\FN@pp@lastfoot{1}\%
\else
\fi
\else
\if\FN@pp@debug
\typeout{not considering footnote serial number #1 (last valid was \FN@pp@lastfoot)}\%
\fi
\fi

\FN@pp@resetfn Set the flag to reset the footnote number; this constructs a chain through the footnote serial numbers at the start of each page
\newcommand{\FN@pp@resetfn}{2}\%
  \gdef\FN@pp@cpage{#2}\%
  \expandafter\gdef\csname FN@pp@next-\FN@pp@prev@foot\endcsname{#1}\%
  \def\FN@pp@prev@foot{#1}\%
  \expandafter\xdef\csname FN@pp@next-\FN@pp@prev@foot\endcsname{\the\@MM}\%
\FN@pp@prev@foot The base of the footnote serial number chain (this element is never looked at; footnote serial 1 must always have number 1)
\newcommand{\FN@pp@prev@foot}{root}

again, avoid confusion; also establish \FN@pp@footnote@aux in the \aux file
\AtBeginDocument{\protected@writeaux{\relax}{\protect\providecommand{\protect\FN@pp@footnote@aux}{2}\{}}%\protect\FN@pp@next@reset\%\ne}
\AtEndDocument{\let\FN@pp@footnote@aux=\FN@pp@footnote@endaux
  \def\FN@pp@lastfoot{-1}\%
  \FN@pp@footnotehintfalse
  \renewcommand{\FN@pp@prev@foot}{root}}
}

22
The footnote analysis command for end document
538 \newcommand{\FN@pp@footnote@endaux}{2}\%
539 \ifnum\FN@pp@lastfoot<#1
540 \if\FN@pp@footnotehint
541 \FN@pp@resetfn@end{#1}{#2}\%
542 \FN@pp@footnotehintfalse
543 \else
544 \gdef@tema{#2}\%
545 \if\tema\FN@pp@cpage
546 \else
547 \FN@pp@resetfn@end{#1}{#2}\%
548 \fi
549 \fi
550 \def\FN@pp@lastfoot{#1}\%
551 ⟨∗PPdebug⟩
552 \else
553 \if\FN@pp@debug
554 \typeout{not considering footnote serial number #1
555 (last valid was \FN@pp@lastfoot)}\%
556 \fi
557 ⟨/PPdebug⟩
558 \fi
559 ⟩

Deal with the .aux file footnote details, at end document
560 \newcommand{\FN@pp@resetfn@end}{2}\%
561 \def@tema{#1}\%
562 \expandafter\ifx\csname FN@pp@next-FN@pp@prev@foot\endcsname@tema%
563 \else
564 \@tempswatrue
565 ⟨∗PPdebug⟩
566 \if\FN@pp@debug
567 \expandafter\ifx\csname FN@pp@next-FN@pp@prev@foot\endcsname\relax
568 \if\FN@pp@lastseq\else
569 \typeout{footnote sequence lost between pages
570 \FN@pp@cpage\space and #2}\%
571 \global\FN@pp@lastseqtrue
572 \fi
573 \else
574 \typeout{footnotes changed between pages \FN@pp@cpage\space and #2:
575 next was \csname FN@pp@next-FN@pp@prev@foot\endcsname, now #1}\%
576 \global\FN@pp@lastseqfalse
577 \fi
578 ⟨/PPdebug⟩
579 \fi
580 ⟨/PPdebug⟩
581 \gdef\FN@pp@prev@foot{#1}\%
582 \gdef\FN@pp@cpage{#2}\%
583 ⟩

\FN@pp@resetfn@end Deal with the .aux file footnote details, at end document
\newcommand{\FN@pp@resetfn@end}{2}\%
\def@tema{#1}\%
\expandafter\ifx\csname FN@pp@next-FN@pp@prev@foot\endcsname@tema%
\else
\@tempswatrue
⟨∗PPdebug⟩
\if\FN@pp@debug
\expandafter\ifx\csname FN@pp@next-FN@pp@prev@foot\endcsname\relax
\if\FN@pp@lastseq\else
\typeout{footnote sequence lost between pages \FN@pp@cpage\space and #2}\%
\global\FN@pp@lastseqtrue
\fi
\else
\typeout{footnotes changed between pages \FN@pp@cpage\space and #2:
next was \csname FN@pp@next-FN@pp@prev@foot\endcsname, now #1}\%
\global\FN@pp@lastseqfalse
\fi
\fi
⟨/PPdebug⟩
\fi
\gdef\FN@pp@prev@foot{#1}\%
\gdef\FN@pp@cpage{#2}\%
⟩

Now, how do we cope with \input documents? We can’t insert anything (the \input package offers \AtBeginInputDocument but there’s no corresponding command for files included by the kernel \input. So we insert a footnote hint at every \clearpage

\clearpage
The business end of the option: a macro to decide on footnote numbers, called from \footnote and \footnotemark (see above).

In case we're also doing multiple option, we now save up the command to write to the .aux file, and mark we've done so

In case we're also doing multiple option, we now save up the command to write to the .aux file, and mark we've done so

Now look at the next element in the chain:

If the chain is broken here, set the next reset point to something (one hopes) infeasibly large... a weak point?

Finally, if we're not doing paragraph footnotes, we redefine \makefntext to take account of the value of \footnotemargin, to impose \footnotelayout, and to make the footnote body text hang, if appropriate.
get the marker so we can measure it:

\setbox\@tempboxa\hbox{%
  \ifdim\footnotemargin>0pt
    \hboxto\footnotemargin{\@makefnmark\hss}%
  \else
    \@makefnmark
  \fi
}\fi

use the width of the box to set up hanging (potentially for more than one paragraph); note that the hanging \parskip and \parindent are set after we’ve executed \leavevmode(!)

\leftmargin\wd\@tempboxa
\rightmargin\z@\linewidth\columnwidth
\advance\linewidth-\leftmargin
\parshape\@ne\leftmargin\linewidth
\footnotesize
\@setpar{{\@par}}%
and finally put the marker in its chosen place:

\leavevmode
\llap{\box\@tempboxa} %
\setpar{({\@par})}%
if we’re hanging, close the hang group
\ifFN@hangfoot
\par\egroup
\fi
}

6 Remaining requirements

We have to insert the code that executes the stable and multiple options. Since stable may suppress the setting of a footnote altogether, we put the multiple
option first, as otherwise we might get isolated superscripted commas that separate footnotes that have otherwise been suppressed.

6.1 The code that executes the multiple option

This (revised) code derives from a suggestion by Alexander Rozhenko (the author of the manyfoot package): the intention is that footmisc and manyfoot should be able to ‘interwork’, in the sense that each would recognise the other’s footnote marks and behave appropriately. The trick is that both \footnote and \footnotemark insert a marker (a cancelling pair of kerns of \multiplefootnotemarker (of opposite signs), which is detected in following \footnote or \footnotemark commands. Note we have to take special precautions to ensure that the kerns are the last things added to the horizontal list by the commands.

\ifFN@multiplefootnote
  \providecommand*{\multiplefootnotemarker}{3sp}
  \providecommand*{\multfootsep}{,}
  \CheckCommand*{\@footnotemark}{%
    \leavevmode
    \ifhmode\edef\@x@sf{\the\spacefactor}\nobreak\fi
    \@makefnmark
    \ifhmode\spacefactor\@x@sf\fi
    \relax
  }
  \renewcommand*{\@footnotemark}{%
    \leavevmode
    \ifhmode
      \edef\@x@sf{\the\spacefactor}\FN@mf@check
      \nobreak
    \fi
    \@makefnmark
    \textsuperscript{\multfootsep}
  }
\fi

if we’re also doing option perpage, write its stuff to the .aux file for it, so the wotsit node doesn’t interfere with our \kern detection.

\ifFN@pp@towrite
  \FN@pp@writetemp
  \FN@pp@towritefalse
  \fi
  \FN@mf@prepare
  \ifhmode\spacefactor\@x@sf\fi
  \relax
}\def\FN@mf@prepare{%
  \kern-\multiplefootnotemarker
  \kern\multiplefootnotemarker\relax
}\def\FN@mf@check{%
  \ifdim\lastkern=\multiplefootnotemarker\relax
  \edef\@x@sf{\the\spacefactor}\relax
  \unkern
  \textsuperscript{\multfootsep}\relax
  \spacefactor\@x@sf\relax
6.2 The code that executes the stable option

The basic idea is to use the ‘original’ code of \footnote (which this package may have hacked around something chronic) only if we’re in typesetting mode (as determined by the state of the \protect command. Otherwise, the command becomes an elaborate multistage ‘gobble’.

\if\BO@stablefootnote
\let\BO@sf@@footnote=\footnote
\def\footnote{\if\protect\@typeset\protect\BO@sf@@footnote\else\expandafter\BO@sf@gobble@opt\fi}
\edef\BO@sf@gobble@opt\noexpand\protect\expandafter\noexpand\csname BO@sf@gobble@opt\endcsname\expandafter\def\csname BO@sf@gobble@opt\endcsname{\@ifnextchar[\BO@sf@gobble@twobracket\@gobble}
\def\BO@sf@gobble@twobracket[#1]#2{}
\fi

Define \BO@sf@gobble@opt as a robust command that gobbles either an optional and a mandatory argument, or just a mandatory one.

\edef\BO@sf@gobble@optonly\noexpand\protect\expandafter\noexpand\csname BO@sf@gobble@optonly\endcsname\expandafter\def\csname BO@sf@gobble@optonly\endcsname{\@ifnextchar[\BO@sf@gobble@bracket{}\@gobble}
\def\BO@sf@gobble@bracket[#1]{}\fi

Now the same for \footnotemark

\if\BO@stablefootnotemark
\let\BO@sf@@footnotemark=\footnotemark
\def\footnotemark{\if\protect\@typeset\protect\BO@sf@@footnotemark\else\expandafter\BO@sf@gobble@optonly\fi}
\edef\BO@sf@gobble@optonly\noexpand\protect\expandafter\noexpand\csname BO@sf@gobble@optonly\endcsname\expandafter\def\csname BO@sf@gobble@optonly\endcsname{\@ifnextchar[\BO@sf@gobble@bracket{}\@gobble}
\def\BO@sf@gobble@bracket{}\fi

\edef\BO@sf@gobble@bracket[#1]{}}
\fi


\setfnsymbol\FN@fnsymbol@lamport

7 Symbol option variants

Lamport’s choice of symbols for \fnsymbol wasn’t entirely “traditional”, so we
(now) provide alternatives. The \setfnsymbol command offers a small number
of choices, and the user may define more still, using the \DefineFNsymbols or
\DefineFNsymbolsTM commands, defined below.

\newcommand{\setfnsymbol}[1]{%
\@bsphack
\@ifundefined{FN@fnsymbol@#1}{}%
\PackageError{footmisc}{Symbol style "#1" not known}%
\@eha
\expandafter\let\expandafter\@fnsymbol\csname
FN@fnsymbol@#1\endcsname
\@esphack}

The default selection is Lamport’s original, as represented in current \LaTEX —
we preserve it in case we need to “get back” to it.
\let\FN@fnsymbol@lamport\@fnsymbol
⟨/package⟩

We need another temp conditional
\newif\if@tempswb
\@tempswbfalse
\if@tempswbtrue
\DefineFNsymbols
\@DefineFNsymbols
\@DefineFNsymbols@
\FN@build@symboldef

The macro \DefineFNsymbols allows the user to define a set of footnote symbols,
to be used with the \setfnsymbol command. Syntax:
\DefineFNsymbols[*]{\(set\ name\)}{\(style\)}{\(symbol\ list\)}

If the optional asterisk is present, the set defined will produce an error if the
symbol number is too large; otherwise it will quietly change to numbering in place
of symbol use (a warning is produced at the end of the document). The set name
is the future argument of \setfnsymbol). The style (default text) gives the style
the symbols are typeset (this is the correct method, but unfortunately not all
symbols, even for Lamport’s original set for \LaTEX \fnsymbol may be expressed
this way in a sufficiently old \LaTEX distribution). The symbol list is a set of objects
to be used when the set is selected.

Example of use:
define a direct replacement for Lamport’s original \fnsymbol command —
\DefineFNsymbols*{lamport}{\(math\)}{\(*\dagger\ddagger\mathsection
\mathparagraph\|\&\)}{\{\dagger\ddagger\}}

Note that doubled-up (and worse — see below) symbols need braces around them.
\newcommand{\DefineFNsymbols}{%
\@ifstar{\@tempsubtrue\@DefineFNsymbols}{}%
\@tempsubfalse\@DefineFNsymbols%
}

\newcommand{\@DefineFNsymbols}[1]{%
\@ifnextchar[{}\@DefineFNsymbols@{#1}{text}{}%
\@DefineFNsymbols@{#1}{text}{}%
Now do the same job for the “modern” way of having both text and maths variants of everything.

\DefineFNsymbolsTM
\@DefineFNsymbolsTM
\FN@build@symboldefTM
\DefineFNsymbolsTM
Note that this version has two variants of every definition, so needs two stopper
codes above.

This is a stripped down (e-Tex only) version of what appears in fixltex2e. If the
command's already defined, we assume it's that version.

Macros to deal with footnote symbols going out of range (when they’re allowed
to – e.g., in the symbol* option).
This is defined in recent \LaTeX releases, but not in (for example) that distributed with the last release of \TeX. Since it’s needed in some symbol set definitions (including Lamport’s) we define it here.

\ifdefined{textbardbl}{%
  \DeclareTextSymbol{\textbardbl}{OMS}{107}%
  \DeclareTextSymbolDefault{\textbardbl}{TS1}}{}%

(This definition comes from the \LaTeX sources.)

\FNsymbol@bringhurst
\FNsymbol@chicago
\FNsymbol@wiley
\FNsymbol@lamport-robust
\FNsymbol@lamport

These macros provide replacement orderings (and symbol sets) for footnote symbols, plus a robust version of the original Lamport set, and an extended version of Lamport’s original

\DefineFNsymbolsTM*{bringhurst}{}
\DefineFNsymbolsTM*{chicago}{}
\DefineFNsymbolsTM*{wiley}{}
\DefineFNsymbolsTM{lamport-robust}{}
8 Other miscellaneous commands

8.1 Footnote references

\footref Syntax: \footref{\textit{label-name}}

One often wishes to refer to a footnote; in some circumstances, \footnotemark just isn’t good enough (for example, inside a minipage, when \footnotemark creates a reference to footnotes outside the minipage).
addresses this problem by making a label reference that actually looks like a \footnotemark. (The command is available in the \texttt{memoir} class, and we therefore \texttt{providecommand} it rather than defining it “outright”.)

\providecommand*{\footref}\[1\]{% 
\begingroup 
\unrestored@protected@xdef\@thefnmark{\ref{#1}}% 
\endgroup 
\@footnotemark % 
}

8.2 Minipage \texttt{\footnotemark}

\texttt{\mpfootnotemark} Syntax: \texttt{\mpfootnotemark\[\langle number\rangle\]}

Here we define \texttt{\mpfootnotemark}, which has the same syntax as \texttt{\footnotemark}, and which applies the semantics of \texttt{\footnotemark} to the minipage footnote series.

\newcommand\mpfootnotemark{% 
@ifnextchar[\@xmpfootnotemark{% 
\stepcounter\@mpfn 
\protected@xdef\@thefnmark{\thempfn} \@footnotemark % 
}%}{% 
\def\@xmpfootnotemark\[#1\]{% 
\begingroup 
\csname c@\@mpfn\endcsname #1\relax 
\unrestored@protected@xdef\@thefnmark{\thempfn} \endgroup 
\@footnotemark % 
}% 
\endgroup 
\@footnotemark % 
}% 
\def\@xmpfootnotemark\[#1\]{% 
\begingroup 
\csname c@\@mpfn\endcsname #1\relax 
\unrestored@protected@xdef\@thefnmark{\thempfn} \endgroup 
\@footnotemark % 
}% 
\endinput

Index

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