

footmisc —
a portmanteau package
for customising footnotes in L^AT_EX*

Robin Fairbairns[†]

May 2, 2004

Support

Some support of this package is available *for unchanged copies of the package* via email to me at the address given in the footnote. Support will remain available at least as long as the address remains valid. I don't guarantee anything, but I will probably look at any bug reports you may submit, and may correct problems reported (either in my code or in yours...).

Copyright statement

Program: `footmisc.dtx`

Copyright 1995 1996 1998 1999 2001–2003 Robin Fairbairns

This program may be redistributed and/or modified under the terms of the LaTeX Project Public License in file `lpp1.txt` (available from CTAN directory `macros/latex/base`; either version 1.2 of the License, or (at your option) any later version).

1 User interface — package options

The `footmisc` package provides several different customisations of the way footnotes are represented in L^AT_EX 2_ε 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 v5.3c, last revised 2004/05/02

[†]University of Cambridge Computer Laboratory, William Gates Building, J. J. Thompson Avenue, Cambridge, CB2 0FD, UK (rf10@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 L^AT_EX 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.

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. The `ragged` option simply sets `\footnotelayout`

to `\raggedright`. (Hint: if you intend to use the `ragged2e` package, load it before `footmisc` — if `footmisc` finds `\RaggedRight` available, it 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 L^AT_EX 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 L^AT_EX 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 L^AT_EX 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. L^AT_EX 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. L^AT_EX’s default set would be defined by the command:

```
\DefineFNSymbols*{lamport}{*\dagger\ddagger\S\P\|}%
{**}{\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`:

bringhurst	* † ‡ § ¶
chicago	* † ‡ § #
wiley	* ** † ‡ § ¶

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 \LaTeX would normally set the footnotes a mere `\skip\footins` distant from the bottom of the text.

There's a further infelicity in \LaTeX 's placing of footnotes of the bottom of pages: if a bottom float appears on a page, \LaTeX 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 ≤ 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:

<code>\hangfootparskip</code>	<code>0.5\baselineskip</code>	The user may redefine these val-
<code>\hangfootparindent</code>	<code>0em</code>	

ues (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, \LaTeX ’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

mumble¹³¹⁴

What the `multiple` option makes of the above is

mumble^{13,14}

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

The option also treats `\footnotemark` in the same way.

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: `\footref` and `\mpfootnotemark`.

When you're in a minipage, `\footnote` numbers run according to the minipage's own footnote counter, and the marks are set in italic letters. However, the numbers used by `\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 L^AT_EX to "get around" it. The command `\mpfootnotemark` gets around this problem in a minipage, by generating footnote marks in the same way as those used by `\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 `\footnotemark` or `\mpfootnotemark`. This is a tedious way of going about things, and doesn't allow representation of all possible forms of footnote mark; `\footref` is a form of reference command that sets the reference as if it were a footnote. The label should be set *within* the argument of the footnote command that is being labelled:

```
... \footnote{Note text\label{fnlabel}}
...
... potato head\footref{fnlabel}
```

2 User interface — interactions with other packages

The `footmisc` package modifies several parts of the L^AT_EX 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:

setspace The `setspace` package modifies the line spacing is calculated in footnotes. `Footmisc` knows about this, and preserves the change. However, you *must* load `setspace` *before* `footmisc`.

hyperref 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.

manyfoot 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 (*package)
   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 pack-
age (still, after all this time, not out of beta status).
4 \providecommand\protected@writeaux{%
5   \protected@write\@auxout
6 }

\l@advance@macro We make the following (\@@dvance@macro) generalisable as follows (the global
@@dvance@macro form isn’t used in this package ... yet):
\@advance@macro
7 \def\l@advance@macro{\@@dvance@macro\edef}
8 \def\@@dvance@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\@advance@macro\l@advance@macro

\footnotemargin Finally, we define the length used by the marginal option, and initialise it as if
we’ve not had the option.
13 \newdimen\footnotemargin
14 \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 L^AT_EX 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.

```

15 \DeclareOption{symbol}{\renewcommand\thefootnote{\fnsymbol{footnote}}}

```

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).

```
16 \DeclareOption{symbol*}{%
17   \renewcommand\thefootnote{\@fnsymbol\c@footnote}%
18   \AtEndOfPackage{\setfnsymbol{lampport*-robust}}%
19 }
```

4.3 The `para` option

The basis of the code for this option comes from `TEXbook`, p.398 ff. (“Dirty Tricks”), though it does (of course) avoid redefining `\` which has some other (somewhat significant) uses in `LATEX`! 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 `TEXbook`, 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 `LATEX`, provided that you reset the footnote counter to make the joins seamless.

`\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.

```
20 \newif\ifFN@para \FN@parafalse
21 \DeclareOption{para}{\ifFN@sidefn
22   \PackageError{footmisc}{Option "\CurrentOption" incompatible with
23     option "side"}%
24   {I shall ignore "\CurrentOption"}%
25   \else
26     \FN@paratrue
27   \fi
28 }
```

4.4 The `side` option

`\ifFN@sidefn` Simply changes the behaviour of `\@footnotetext`; incompatible with paragraph footnotes.

```
29 \newif\ifFN@sidefn \FN@sidefnfalse
30 \DeclareOption{side}{\ifFN@para
31   \PackageError{footmisc}{Option "\CurrentOption" incompatible with
32     option "para"}%
33   {I shall ignore "\CurrentOption"}%
34   \else
35     \FN@sidefntrue
36   \fi
37 }
```

4.5 The ragged option

`\footnotelayout` A very simple option that merely changes the definition of one macro. Note detection of the presence of the `ragged2e` package.

```
38 \let\footnotelayout\relax
39 \DeclareOption{ragged}{%
40   \@ifundefined{RaggedRight}{%
41     {\def\footnotelayout{\linepenalty50 \raggedright}}%
42     {\def\footnotelayout{\linepenalty50 \RaggedRight}}%
43 }
```

4.6 The 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).

```
44 \newif\ifFN@perpage
45 \FN@perpagefalse
46 \DeclareOption{perpage}{%
47   \FN@perpagetrue
48 }
```

4.7 The 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 `.ins` file to generate a version of the package that includes the option, if you feel you need it.

```
49 (*PPdebug)
50 \newif\ifFN@pp@debug \FN@pp@debugfalse
51 \DeclareOption{PPdebug}{\FN@pp@debugtrue}
52 \PPdebug)
```

4.8 The bottom option

`\ifFN@bottom` All this needs to do is to set a flag to say that it should happen

```
53 \newif\ifFN@bottom \FN@bottomfalse
54 \DeclareOption{bottom}{%
55   \FN@bottomtrue
56 }
```

4.9 The marginal option

Again, the processing of the option is pretty trivial:

```
57 \DeclareOption{marginal}{%
58   \footnotemargin-0.8em\relax
59 }
```

4.10 The flushmargin option

Again, the processing of the option is pretty trivial:

```
60 \DeclareOption{flushmargin}{%
```

```
61 \footnotemarginOpt\relax
62 }
```

4.11 The hang option

`\ifFN@hangfoot` We need a switch, since `\@makefn` needs to be patched.

```
63 \newif\ifFN@hangfoot \FN@hangfootfalse
64 \DeclareOption{hang}{%
65 \FN@hangfoottrue
66 }
```

`\hangfootparskip` `\hangfootparindent` Layout parameters for hanging footnotes; `\hangfootparskip` and `\hangfootparindent` are (respectively) values to use for `\parskip` and `\parindent` when in hanging footnotes.

```
67 \newcommand*\hangfootparskip{0.5\baselineskip}
68 \newcommand*\hangfootparindent{0em}%
```

4.12 The norule option

Pretty simple too...

```
69 \DeclareOption{norule}{%
70 \renewcommand\footnoterule{}%
71 \advance\skip\footins 4\p@\@plus2\p@\relax
72 }
```

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 L^AT_EX only uses inserts for footnotes, so that if any insert is going to be split, it's going to be a footnote.

```
73 \DeclareOption{splitrule}{%
74 \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...

```
75 \let\pagefootnoterule\footnoterule
76 \let\mpfootnoterule\footnoterule
77 \def\splitfootnoterule{\kern-3\p@ \hrule \kern2.6\p@}
```

Now redefine `\footnoterule` to distinguish the three situations.

```
78 \def\footnoterule{\relax
79 \ifx \@listdepth\@mplistdepth
```

In a minipage

```
80 \mpfootnoterule
81 \else
82 \ifnum\split@prev=\z@
```

Normal footnote on a regular page

```
83 \pagefootnoterule
84 \else
```

```

      Second part of a split footnote
85     \splitfootnoterule
86     \fi
      Remember a split for next page
87     \xdef\split@prev{\the\insertpenalties}%
88     \fi
89   }%
90 }

```

`\ifFN@stablefootnote` 4.14 The stable option

Simply set a flag: the code of this gets executed at the very end of the package.

```

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

```

4.15 The multiple option

`\ifFN@multiplefootnote` Again, simply set a flag, for code that gets executed at the very very very end of the package.

```

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

```

4.16 The start of the endgame

Exercise the options that the user has requested...

```

95 \ProcessOptions

```

5 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 `para` or `bottom` has been invoked, we do.

```

96 \let \if@tempswa \ifFN@bottom
97 \ifFN@para \@tempwatrue \fi
98 \if@tempswa

```

... so we've patching to do.

First, we ensure that `\@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.)

```

99 \@ifl@t@r\fmtversion{2003/12/01}{%
100 \CheckCommand*\@makecol{\ifvoid \footins

```

```

101     \setbox\@outputbox \box\@cclv
102 \else
103     \setbox\@outputbox \vbox{%
104         \boxmaxdepth\@maxdepth
105         \@tempdima\dp\@cclv
106         \unvbox\@cclv
107         \vskip \skip\footins
108         \color@begingroup
109             \normalcolor\footnoterule
110             \unvbox\footins
111         \color@endgroup
112     }%
113 \fi
114 \let \@elt \relax
115 \xdef\@freelist{\@freelist\@midlist}%
116 \global\let\@midlist\@empty
117 \@combinefloats
118 \ifvbox\@kludgeins
119     \@makespecialcolbox
120 \else
121     \setbox\@outputbox \vbox to\@colht{%
122         \@texttop \dimen@dp\@outputbox
123         \unvbox\@outputbox
124         \vskip -\dimen@\@textbottom
125     }%
126 \fi
127 \global\maxdepth\@maxdepth
128 }
129 }{%
130 \@ifl@t@r\fmtversion{1999/12/01}{%
131     \CheckCommand*\@makecol{\ifvoid \footins
132         \setbox\@outputbox \box\@cclv
133     \else
134         \setbox\@outputbox \vbox{%
135             \boxmaxdepth\@maxdepth
136             \@tempdima\dp\@cclv
137             \unvbox\@cclv
138             \vskip \skip\footins
139             \color@begingroup
140                 \normalcolor\footnoterule
141                 \unvbox\footins
142             \color@endgroup
143         }%
144     \fi
145     \xdef\@freelist{\@freelist\@midlist}%
146     \global\let\@midlist\@empty
147     \@combinefloats
148     \ifvbox\@kludgeins
149         \@makespecialcolbox
150     \else
151         \setbox\@outputbox \vbox to\@colht{%
152             \@texttop \dimen@dp\@outputbox
153             \unvbox\@outputbox
154             \vskip -\dimen@\@textbottom

```

```

155     }%
156     \fi
157     \global\maxdepth\@maxdepth
158   }
159 }{%
160 \CheckCommand*\@makecol{\ifvoid \footins
161   \setbox\@outputbox \box\@cclv
162   \else
163     \setbox\@outputbox \vbox{%
164       \boxmaxdepth\@maxdepth
165       \unvbox\@cclv
166       \vskip \skip\footins
167       \color@begingroup
168         \normalcolor\footnoterule
169         \unvbox\footins
170       \color@endgroup
171     }%
172   \fi
173   \xdef\@freelist{\@freelist\@midlist}%
174   \global\let\@midlist\@empty
175   \@combinefloats
176   \ifvbox\@kludgeins
177     \@makespecialcolbox
178   \else
179     \setbox\@outputbox \vbox to\@colht{%
180       \@texttop \dimen@\dp\@outputbox
181       \unvbox\@outputbox
182       \vskip -\dimen@\@textbottom
183     }%
184   \fi
185   \global\maxdepth\@maxdepth
186 }
187 }
188 }

```

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

```
189 \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`.

```

190 \FN@emptoken{%
191   \toks@\expandafter{\the\toks@
192     \vskip\skip\footins
193     \color@begingroup
194       \normalcolor\footnoterule
195     \global\setbox\FN@tempboxc\vbox{\makefootnoteparagraph}%
196     \unvbox\FN@tempboxc
197   \color@endgroup
198 }%

```

```
199 }%
```

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

```
200 \else
201   \FN@temptoken{%
202     \toks@expandafter{\the\toks@
203       \vskip\skip\footins
204       \color@begingroup
205       \normalcolor\footnoterule
206       \unvbox\footins
207       \color@endgroup
208     }%
209   }%
210 \fi
```

Now we start building up the revised version of `\@makecol`. The definition starts out in `\toks@`; first the bottom version:

```
211 \ifFN@bottom
212   \toks@{\setbox\@outputbox \box\@cclv
213     \xdef\@freelist{\@freelist\@midlist}%
214     \global\let\@midlist\@empty
215     \@combinefloats
216     \ifvoid\footins
217     \else
218       \setbox\@outputbox \vbox\bgroup
219       \boxmaxdepth\@maxdepth
220       \unvbox\@outputbox
221       \fill\relax
222     }
223   \the\FN@temptoken
224   \toks@\expandafter{\the\toks@\egroup\fi}
```

Not putting stuff at the bottom: footnotes are placed using the kernel's algorithm.

```
225 \else
226   \toks@{\ifvoid\footins
227     \setbox\@outputbox\box\@cclv
228     \else
229       \setbox\@outputbox \vbox\bgroup
230       \boxmaxdepth\@maxdepth
231       \unvbox\@cclv
232     }
233   \the\FN@temptoken
```

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@`.

```
234   \toks@\expandafter{\the\toks@
235     \egroup
236     \fi
237     \xdef\@freelist{\@freelist\@midlist}%
238     \global\let\@midlist\@empty
239     \@combinefloats
240   }%
241 \fi
```

Finally, create the new definition from the resulting object with the remainder of the original `\@makecol` tagged on at the end.

```

242 \toks@\expandafter{\the\toks@
243 \ifvbox\@kludgeins
244 \@makespecialcolbox
245 \else
246 \setbox\@outputbox \vbox to\@colht{%
247 \@texttop \dimen@\dp\@outputbox
248 \unvbox\@outputbox
249 \vskip -\dimen@\@textbottom
250 }%
251 \fi
252 \global\maxdepth\@maxdepth
253 }
254 \edef\@makecol{\the\toks@}

```

All of the above occurred conditionally on the ‘or’ of `\ifFN@para` and `\ifFN@bottom`, so we now close the conditional.

```
255 \fi
```

5.2 The requirements of `\@footnotetext`

`\ifFN@setspace` 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:

```

256 \newif\ifFN@setspace
257 \@ifpackageloaded{setspace}{%
258 \FN@setspace>true
259 \CheckCommand\@footnotetext[1]{%
260 \insert\footins{%
261 \def\baselinestretch {\setspace@singlespace}%
262 \reset@font\footnotesize
263 \interlinepenalty\interfootnotelinepenalty
264 \splittopskip\footnotesep
265 \splitmaxdepth \dp\strutbox
266 \floatingpenalty\@MM
267 \hsize\columnwidth
268 \@parboxrestore
269 \protected@edef\@currentlabel{%
270 \csname p@footnote\endcsname\@thefnmark
271 }%
272 \color@begingroup
273 \@makefntext{\rule\z@\footnotesep
274 \ignorespaces#1\@finalstrut\strutbox
275 }%
276 \color@endgroup
277 }%
278 }%
279 }{%
280 \FN@setspace>false
281 \@ifl@t@r\fmtversion{1997/12/01}{%
282 \CheckCommand\@footnotetext[1]{%
283 \insert\footins{%

```

```

284     \reset@font\footnotesize
285     \interlinepenalty\interfootnotelinepenalty
286     \splittopskip\footnotesep
287     \splitmaxdepth \dp\strutbox
288     \floatingpenalty\@MM
289     \hsize\columnwidth
290     \@parboxrestore
291     \protected@edef\@currentlabel{%
292       \csname p@footnote\endcsname\@thefnmark
293     }%
294     \color@begingroup
295     \@makefntext{\rule\z@\footnotesep
296       \ignorespaces#1\@finalstrut\strutbox
297     }%
298     \color@endgroup
299   }%
300 }%
301 }{%
302   \CheckCommand\@footnotetext[1]{%
303     \insert\footins{%
304       \reset@font\footnotesize
305       \interlinepenalty\interfootnotelinepenalty
306       \splittopskip\footnotesep
307       \splitmaxdepth \dp\strutbox
308       \floatingpenalty\@MM
309       \hsize\columnwidth
310       \@parboxrestore
311       \protected@edef\@currentlabel
312         {\csname p@footnote\endcsname\@thefnmark}%
313       \color@begingroup
314         \@makefntext{\rule\z@\footnotesep
315           \ignorespaces#1\@finalstrut\strutbox
316       }
317     \color@endgroup
318   }%
319 }%
320 }%
321 }

```

(The 1997/06/01 L^AT_EX source really doesn't have a comment mark after that closing brace.)

There's substantial patching to be done if we're doing paragraph footnotes:

```

322 \ifFN@para
323   \renewcommand\@footnotetext[1]{%
324     \insert\footins{%
325       insert compatibility code with setspace.sty if necessary
326       \iffN@setspace
327         \def\baselinestretch {\setspace@singlespace}%
328       \fi
329       \reset@font\footnotesize
330       \interlinepenalty\interfootnotelinepenalty
331       \splittopskip\footnotesep
332       \splitmaxdepth \dp\strutbox

```

```

332     \floatingpenalty\@MM
333     \hsize\columnwidth
334     \@parboxrestore
335     \protected@edef\@currentlabel{\csname p@footnote\endcsname\@thefnmark}%
336     \color@begingroup

```

We set the paragraph in an `\hbox` and apply the fudge factor here:

```

337     \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.

```

338         \@makefntext{\ignorespaces#1\strut

```

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

```

339             \penalty-10\relax
340             \hskip\footglue
341         }% end of \@makefntext parameter
342     }% end of \hbox
343     \dp\FN@tempboxa=0pt\ht\FN@tempboxa=\fudgefactor\wd\FN@tempboxa
344     \box\FN@tempboxa
345     \color@endgroup
346 }%
347 \FN@mfb@prepare
348 }

```

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

```

349 \else
350     \ifFN@sidefn
351     \renewcommand\@footnotetext[1]{%
352         \marginpar{%

```

insert compatibility code with `setspace.sty` if necessary

```

353         \ifFN@setspace
354         \def\baselinestretch {\setspace@singlespace}%
355         \fi
356         \reset@font\footnotesize
357         \protected@edef\@currentlabel{%
358             \csname p@footnote\endcsname\@thefnmark
359         }%
360         \color@begingroup
361         \@makefntext{%
362             \ignorespaces#1%
363         }%
364         \color@endgroup
365     }%
366     \FN@mfb@prepare
367 }%
368 \else
369     \renewcommand\@footnotetext[1]{%
370         \insert\footins{%

```

insert compatibility code with `setspace.sty` if necessary

```

371     \ifFN@setspace
372     \def\baselinestretch {\setspace@singlespace}%
373     \fi
374     \reset@font\footnotesize
375     \interlinepenalty\interfootnotelinepenalty
376     \splittopskip\footnotesep
377     \splitmaxdepth \dp\strutbox
378     \floatingpenalty\@MM
379     \hsize\columnwidth
380     \@parboxrestore
381     \protected@edef\@currentlabel{%
382       \csname p@footnote\endcsname\@thefnmark
383     }%
384     \color@begingroup
385     \@makefnmark{%
386       \rule{z@\footnotesep
387         \ignorespaces#1\@finalstrut\strutbox
388       }%
389     \color@endgroup
390   }%
391   \FN@mfb@prepare
392 }%
393 \fi
394 \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.

```
395 \ifFN@para
```

```

\FN@tempboxa We need some temporary boxes, and LATEX only defines one
\FN@tempboxb 396 \let\FN@tempboxa\@tempboxa
\FN@tempboxb 397 \newbox\FN@tempboxb
398 \newbox\FN@tempboxc

```

```
\footglue A direct crib from the TEXbook:
```

```
399 \newskip\footglue \footglue=1em plus.3em minus.3em
```

```
\@makefnmark 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 `\@makefnmark` being executed in restricted vertical mode, which results in its `\hbox` ending up in a line of its own.

```

400 \long\def\@makefnmark#1{\leavevmode
401   \@makefnmark\nobreak
402   \hskip.5em\relax#1%
403 }

```

`\footnotebaselineskip` We need to record a value for the baseline skip when in footnotes:

```
404 \newdimen\footnotebaselineskip
405 {%
406   \footnotesize
407   \global
408   \footnotebaselineskip=\normalbaselineskip
409 }
```

`\fudgefactor` Now we derive a fudge factor from the `baselineskip` we've just established:

```
410 \@tempdima=\footnotebaselineskip \multiply\@tempdima by 1024
411 \divide \@tempdima by \columnwidth \multiply\@tempdima by 64
412 \xdef\fudgefactor{\strip@pt\@tempdima }%
```

`\makefootnoteparagraph` For use in the output routine

```
413 \long\def\makefootnoteparagraph{\unvbox\footins \makeboxofhboxes
414   \setbox\FN@tempboxa=\hbox{\unhbox\FN@tempboxa \removehboxes}
```

Now we are ready to set the paragraph:

```
415   \hsize\columnwidth
416   \@parboxrestore
417   \baselineskip=\footnotebaselineskip
418   \noindent
419   \rule{\z@}{\footnotesep}%
420   \unhbox\FN@tempboxa\par
421 }
```

`\makeboxofhboxes` Support code for `\makefootnoteparagraph`

```
\removehboxes 422 \def\makeboxofhboxes{\setbox\FN@tempboxa=\hbox{}}%
423   \loop
424     \setbox\FN@tempboxb=\lastbox
425     \ifhbox\FN@tempboxb
426       \setbox\FN@tempboxa=\hbox{\box\FN@tempboxb\unhbox\FN@tempboxa}%
427     \repeat
428   }
429 \def\removehboxes{\setbox\FN@tempboxa=\lastbox
430   \ifhbox
431     \FN@tempboxa{\removehboxes}%
432   \unhbox\FN@tempboxa
433   \fi
434 }
435 \fi
```

5.4 The other footnote commands

`\ifFN@pp@footnotehint` A conditional needed by the perpage code: must be defined outside the *perpage* conditional

```
436 \newif\ifFN@pp@footnotehint
```

`\c@pp@next@reset` Counter used to store information about the next reset of the footnote number, in perpage mode.

```
437 \newcounter{pp@next@reset}%
```

`\ifFN@pp@towrite` A conditional that mediates the interaction between the `perpage` option and the `multiple` option.

```
438 \newif\ifFN@pp@towrite
439 \FN@pp@towritefalse
```

`\ifFN@pp@lastseq` We prevent endless processions of diagnostics ‘footnote sequence lost’ by using this conditional (again, needs to be defined outside the `perpage` conditional:

```
440 (*PPdebug)
441 \newif\ifFN@pp@lastseq
442 \global\FN@pp@lastseqfalse
443 \PPdebug)
```

Now, do we need to patch `\footnote` for per-page footnotes?

```
444 \ifFN@perpage
445   \CheckCommand*\footnote{\@ifnextchar [%]
446     \@xfootnote
447     {%
448       \stepcounter\@mpfn \protected@xdef\@thefnmark{\thempfn}%
449       \@footnotemark \@footnotetext
450     }%
451   }
452 \renewcommand*\footnote{\@ifnextchar [%]
453   \@xfootnote
454   {%
455     \stepcounter\@mpfn \protected@xdef\@thefnmark{\thempfn}%
456     \FN@pp@footnote\@footnotemark
```

In case that we’re *not* running `multiple` option, `\@footnotemark` *won’t* have written details to the `.aux` file, so do it now:

```
457   \ifFN@pp@towrite
458     \FN@pp@writetemp
459     \FN@pp@towritefalse
460   \fi
461   \@footnotetext
462 }%
463 }
```

And the analagous change for `\footnotemark`

```
464 \CheckCommand*\footnotemark{%
465   \@ifnextchar [%]
466   \@xfootnotemark
467   {%
468     \stepcounter{footnote}%
469     \protected@xdef\@thefnmark{\thefootnote}%
470     \@footnotemark
471   }%
472 }
473 \renewcommand*\footnotemark{%
474   \@ifnextchar [%]
475   \@xfootnotemark
476   {%
477     \stepcounter{footnote}%
478     \protected@xdef\@thefnmark{\thefootnote}%
479     \FN@pp@footnote\@footnotemark
```

again, tidy up if we're not doing multiple option

```
480     \ifFN@pp@towrite
481         \FN@pp@writetemp
482         \FN@pp@towritefalse
483     \fi
484 }%
485 }
```

`\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

```
486 \gdef\FN@pp@initial@stab{\@addtoreset{footnote}{page}}
487 \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.

```
488 \newcounter{fnserial}
```

`\FN@pp@cpage` The package requires a “knowledge” of the current page number. It's kept in `\FN@pp@cpage`

```
489 \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.

```
490 \FN@pp@footnotehinttrue
491 \newcommand{\footnotehint}{%
492     \setcounter{footnote}{0}%
493     \protected@writeaux\relax{\protect\FN@pp@footnotehinttrue}%
494     \@tempcnta\c@fnserial
495     \advance\@tempcnta\@ne
496     \global\c@pp@next@reset\@tempcnta
497 }
498 \AtBeginDocument{\protected@writeaux\relax{%
499     \protect\providecommand{\protect\FN@pp@footnotehinttrue}{}%
500 }%
501 }
```

`\FN@pp@lastfoot` Dummy value for the number of the last footnote we came across.

```
502 \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

```
503 \newcommand{\FN@pp@footnote@aux}[2]{%
504     \ifnum\FN@pp@lastfoot<#1
505         \ifFN@pp@footnotehint
506             \FN@pp@resetfn{#1}{#2}%
507             \FN@pp@footnotehintfalse
```

```

508     \else
509         \gdef\@tempa{#2}%
510         \ifx\@tempa\FN@pp@cpage
511             \else
512                 \FN@pp@resetfn{#1}{#2}%
513             \fi
514         \fi
515         \def\FN@pp@lastfoot{#1}%
516 < *PPdebug>
517     \else
518         \ifFN@pp@debug
519             \typeout{not considering footnote serial number #1
520                 (last valid was \FN@pp@lastfoot)}%
521         \fi
522 < /PPdebug>
523     \fi

```

Since we've now had at least one item of footnote information from a .aux file, we can't allow footnote resetting per page.

```

524     \global\let\FN@pp@initial@stab\relax
525 }

```

`\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

```

526 \newcommand{\FN@pp@resetfn}[2]{%
527     \gdef\FN@pp@cpage{#2}%
528     \expandafter\gdef
529         \csname FN@pp@next-\FN@pp@prev@foot\endcsname{#1}%
530     \def\FN@pp@prev@foot{#1}%
531     \expandafter\xdef
532         \csname FN@pp@next-\FN@pp@prev@foot\endcsname{\the\@MM}%
533 }

```

`\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)

```

534 \newcommand{\FN@pp@prev@foot}{root}

```

again, avoid confusion; also establish `\FN@pp@footnote@aux` in the .aux file

```

535 \AtBeginDocument{\protected@writeaux\relax{%
536     \protect\providecommand{\protect\FN@pp@footnote@aux}[2]{}}%
537     \c@pp@next@reset@one
538 }

```

At end document, establish a new `\FN@pp@footnote@aux` which checks whether numbers have changed during the scan of the .aux files for changed labels and the like

```

539 \AtEndDocument{\let\FN@pp@footnote@aux\FN@pp@footnote@endaux
540     \def\FN@pp@lastfoot{-1}%
541     \FN@pp@footnotehintfalse
542     \renewcommand{\FN@pp@prev@foot}{root}
543 }

```

`\FN@pp@footnote@endaux` The footnote analysis command for end document

```

544 \newcommand{\FN@pp@footnote@endaux}[2]{%

```

```

545 \ifnum\FN@pp@lastfoot<#1
546 \ifFN@pp@footnotehint
547 \FN@pp@resetfn@end{#1}{#2}%
548 \FN@pp@footnotehintfalse
549 \else
550 \gdef\@tempa{#2}%
551 \ifx\@tempa\FN@pp@cpage
552 \else
553 \FN@pp@resetfn@end{#1}{#2}%
554 \fi
555 \fi
556 \def\FN@pp@lastfoot{#1}%
557 (*PPdebug)
558 \else
559 \ifFN@pp@debug
560 \typeout{not considering footnote serial number #1
561 (last valid was \FN@pp@lastfoot)}%
562 \fi
563 (/PPdebug)
564 \fi
565 }

```

`\FN@pp@resetfn@end` Deal with the .aux file footnote details, at end document

```

566 \newcommand{\FN@pp@resetfn@end}[2]{%
567 \def\@tempa{#1}%
568 \expandafter\ifx\csname FN@pp@next-\FN@pp@prev@foot\endcsname\@tempa%
569 \else
570 \@tempswatruue
571 (*PPdebug)
572 \ifFN@pp@debug
573 \expandafter\ifx\csname FN@pp@next-\FN@pp@prev@foot\endcsname\relax
574 \ifFN@pp@lastseq\else
575 \typeout{footnote sequence lost between pages
576 \FN@pp@cpage\space and #2}%
577 \global\FN@pp@lastseqtrue
578 \fi
579 \else
580 \typeout{footnotes changed between pages \FN@pp@cpage\space and #2:
581 next was \csname FN@pp@next-\FN@pp@prev@foot\endcsname, now #1}%
582 \global\FN@pp@lastseqfalse
583 \fi
584 \fi
585 (/PPdebug)
586 \fi
587 \gdef\FN@pp@prev@foot{#1}%
588 \gdef\FN@pp@cpage{#2}%
589 }

```

`\clearpage` Now, how do we cope with `\included` documents? We can't insert anything (the `\FN@pp@@clearpage` `includex` package offers `\AtBeginIncludedDocument` but there's no corresponding command for files included by the kernel `\include`. So we insert a footnote hint at every `\clearpage`

```

590 \let\FN@pp@@clearpage\clearpage
591 \renewcommand{\clearpage}{\footnotehint\FN@pp@@clearpage}

```

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

```

592 \def\FN@pp@footnote{%
593   \if@minipage\else
594     \global\advance\c@@fnserial\@ne
595     \if@filesw
      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
596     \xdef\FN@pp@writetemp{%
597       \noexpand\protected@writeaux\relax{%
598         \string\FN@pp@footnote@aux
599         {\the\c@@fnserial}{\noexpand\thepage}}%
600     }%
601   }%
602   \FN@pp@towritetrue
603 \fi
604 \ifnum\c@pp@next@reset>\c@@fnserial
605 \else
606   \global\expandafter\csname c@\mpfn\endcsname\@ne
607   \protected@xdef\@thefnmark{\thempfn}%

```

Now look at the next element in the chain:

```

608   \expandafter\let\expandafter\@tempa
609   \csname FN@pp@next-\number\c@pp@next@reset\endcsname

```

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

```

610   \ifx\@tempa\relax
611     \global\c@pp@next@reset\@MM
612   \else
613     \global\c@pp@next@reset\@tempa
614   \fi
615 \fi
616 \fi
617 }

```

End of code loaded when `perpage` option is given

```
618 \fi
```

Finally, if we're not doing paragraph footnotes, we redefine `\@makefnstext` to take account of the value of `\footnotemargin`, to impose `\footnotelayout`, and to make the footnote body text hang, if appropriate.

```

619 \ifFN@para
620 \else
      hanging footnote version:
621   \long\def\@makefnstext#1{%
622     \ifFN@hangfoot
623       \bgroup
      get the marker so we can measure it:
624       \setbox\@tempboxa\hbox{%
625         \ifdim\footnotemargin>0pt
626           \hb@xt@\footnotemargin{\@makefnmark\hss}}%
627       \else

```

```

628         \@makefnmark
629         \fi
630     }%
    use the width of the box to set up hanging (potentially for more than one
    paragraph)
631     \leftmargin\wd\@tempboxa
632     \rightmargin\z@
633     \linewidth \columnwidth
634     \advance \linewidth -\leftmargin
635     \parshape \@ne \leftmargin \linewidth
636     \footnotesize
637     \parskip\hangfootparskip\relax
638     \parindent\hangfootparindent\relax
    stop the \parshape being overwritten:
639     \@setpar{\@par}}%
    and finally put the marker in its chosen place:
640     \leavevmode
641     \llap{\box\@tempboxa}%
642     \else
    ordinary (non-hanging) footnote version:
643     \parindent1em
644     \noindent
645     \ifdim\footnotemargin>\z@
646         \hb@xt@ \footnotemargin{\hss\@makefnmark}%
647     \else
648         \ifdim\footnotemargin=\z@
649             \llap{\@makefnmark}%
650         \else
651             \llap{\hb@xt@ -\footnotemargin{\@makefnmark\hss}}%
652         \fi
653     \fi
654     \fi
655     \footnotelayout#1%
    if we're hanging, close the hang group
656     \ifFN@hangfoot
657         \par\egroup
658     \fi
659 }
660 \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

`\multiplefootnotemark` This (revised) code derives from a suggestion by Alexander Rozhenko (the author of the *manyfoot* package): the intention is that *footmisc* and *many-foot* 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 `\FN@mf@prepare` `\footnote` and `\footnotemark` insert a marker (a cancelling pair of kerns of `\multiplefootnotemark` (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.

```

661 \ifFN@multiplefootnote
662   \providecommand*\multiplefootnotemark}{3sp}
663   \providecommand*\multfootsep}{,}
664   \CheckCommand*\@footnotemark{%
665     \leavevmode
666     \ifhmode\edef\@x@sf{\the\spacefactor}\nobreak\fi
667     \@makefnmark
668     \ifhmode\spacefactor\@x@sf\fi
669     \relax
670   }
671   \renewcommand*\@footnotemark{%
672     \leavevmode
673     \ifhmode
674       \edef\@x@sf{\the\spacefactor}%
675       \FN@mf@check
676       \nobreak
677       \fi
678     \@makefnmark

```

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.

```

679   \ifFN@pp@towrite
680     \FN@pp@writetemp
681     \FN@pp@towritefalse
682   \fi
683   \FN@mf@prepare
684   \ifhmode\spacefactor\@x@sf\fi
685   \relax
686 }
687 \def\FN@mf@prepare{%
688   \kern-\multiplefootnotemark
689   \kern\multiplefootnotemark\relax
690 }
691 \def\FN@mf@check{%
692   \ifdim\lastkern=\multiplefootnotemark\relax
693     \edef\@x@sf{\the\spacefactor}%
694     \unkern
695     \textsuperscript{\multfootsep}%
696     \spacefactor\@x@sf\relax
697   \fi
698 }

```

If we’re not doing multiple, just create an empty `\FN@mf@prepare`

```

699 \else
700 \let\FN@mf@prepare\relax
701 \fi

```

6.2 The code that executes the stable option

`\ifFN@stablefootnote` 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’.

```

702 \ifFN@stablefootnote
703 \let\FN@sf@@footnote\footnote
704 \def\footnote{\ifx\protect\@typeset@protect
705   \expandafter\FN@sf@@footnote
706   \else
707   \expandafter\FN@sf@gobble@opt
708   \fi
709 }

```

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

```

710 \edef\FN@sf@gobble@opt{\noexpand\protect
711   \expandafter\noexpand\csname FN@sf@gobble@opt \endcsname}
712 \expandafter\def\csname FN@sf@gobble@opt \endcsname{%
713   \@ifnextchar [%]
714     \FN@sf@gobble@twobrace
715     \@gobble
716 }
717 \def\FN@sf@gobble@twobrace[#1]#2{}

```

`\FN@sf@@footnotemark` Now the same for `\footnotemark`

```

\FN@sf@gobble@optonly 718 \let\FN@sf@@footnotemark\footnotemark
\FN@sf@gobble@bracket 719 \def\footnotemark{\ifx\protect\@typeset@protect
720   \expandafter\FN@sf@@footnotemark
721   \else
722   \expandafter\FN@sf@gobble@optonly
723   \fi
724 }
725 \edef\FN@sf@gobble@optonly{\noexpand\protect
726   \expandafter\noexpand\csname FN@sf@gobble@optonly \endcsname}
727 \expandafter\def\csname FN@sf@gobble@optonly \endcsname{%
728   \@ifnextchar [%]
729     \FN@sf@gobble@bracket
730     {}%
731 }
732 \def\FN@sf@gobble@bracket[#1]{}
733 \fi

```

7 Symbol option variants

`\setfnsymbol`
`\FN@fnsymbol@lamport`

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 (perhaps there's a future in letting the user loose here ... can't think of a syntax just now).

```

734 \newcommand\setfnsymbol[1]{%
735   \@bsphack
736   \ifundefined{FN@fnsymbol@#1}%
737   {%
738     \PackageError{footmisc}{Symbol style "#1" not known}%
739     \@eha
740   }{%
741     \expandafter\let\expandafter\@fnsymbol\csname
742       FN@fnsymbol@#1\endcsname
743   }%
744   \@esphack
745 }
746 \let\FN@fnsymbol@lamport\@fnsymbol
747 \</package>

```

```

\if@tempwb We need another temp conditional
\@tempwbfalse 748 \newif\if@tempwb
\@tempwbtrue
\DefineFnsymbols The macro \DefineFnsymbols allows the user to define a set of footnote symbols,
\@DefineFnsymbols to be used with the \setfnsymbol command. Syntax:
\@DefineFnsymbols@ \DefineFnsymbols[*]{\set name}{\style}{\symbol list}
\FN@build@symboldef 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 `math`) gives the style the symbols are typeset (in principle, the *correct* method is to use `text` more, but not even Lamport's original set for L^AT_EX `\fnsymbol` may be expressed this way. The symbol list is a set of objects to be used when the set is selected.

Example of use – define a replacement for Lamport's original `\fnsymbol` command:

```

\DefineFnsymbols*{lamport}{*\dagger\ddagger\mathsection
\mathparagraph\|{**}{\dagger\dagger}{\ddagger\ddagger}}
}

```

Note that doubled-up symbols need braces around them.

```

749 \newcommand{\DefineFnsymbols}{%
750   \ifstar{\@tempwbtrue\@DefineFnsymbols}%
751   {\@tempwbfalse\@DefineFnsymbols}%
752 }
753 \newcommand{\@DefineFnsymbols}[1]{%
754   \@ifnextchar[% ]
755   {\@DefineFnsymbols@{#1}}{\@DefineFnsymbols@{#1}[math]}%
756 }
757 \def\@DefineFnsymbols@#1[#2]#3{%
758   \expandafter\ifx\csname FN@fnsymbol@#1\endcsname\relax
759   \PackageInfo{footmisc}{Declaring symbol style #1}%
760   \else
761   \PackageWarning{footmisc}{Redeclaring symbol style #1}%
762   \fi
763   \toks@{ }%

```

```

764 \def\@tempb{\end}%
765 \FN@build@symboldef#3\end
766 \def\@tempc{math}%
767 \def\@tempd{#2}%
768 \expandafter\xdef\csname FN@fnsymbol@#1\endcsname##1{%
769   \ifx\@tempc\@tempd
770     \noexpand\ensuremath
771   \else
772     \noexpand\nfss@text
773   \fi
774   {%
775     \noexpand\ifcase##1%
776     \the\toks@
777     \noexpand\else
778     \if@tempswb
779       \noexpand\@ctrerr
780     \else
781       \noexpand\@arabic##1\noexpand\FN@orange##1%
782     \fi
783     \noexpand\fi
784   }%
785 }%
786 }
787 \def\FN@build@symboldef#1{%
788   \def\@tempa{#1}%
789   \ifx\@tempa\@tempb
790   \else
791     \toks@\expandafter{\the\toks@\or#1}%
792     \expandafter\FN@build@symboldef
793   \fi
794 }

```

`\FN@orange` Macros to deal with footnote symbols going out of range (when they're allowed to—e.g., in the `symbol*` option).

```

\@diagnose@fnsymbol@orange 795 \def\FN@orange#1{%
796   \@bsphack
797   \PackageInfo{footmisc}{Footnote number \number#1 out of range}%
798   \protect\@fnsymbol@orange
799   \@esphack
800 }
801 \global\let\@diagnose@fnsymbol@orange\relax
802 \AtEndDocument{\@diagnose@fnsymbol@orange}
803 \def\@fnsymbol@orange{%
804   \gdef\@diagnose@fnsymbol@orange{%
805     \PackageWarningNoLine{footmisc}{Some footnote number(s)
806       were out of range
807       \MessageBreak
808       see log for details%
809     }%
810   }%
811 }

```

`\FN@fnsymbol@bringhurst` These macros provide replacement orderings (and symbol sets) for footnote symbols, plus a robust version of the original Lamport set, and an extended version

`\FN@fnsymbol@chicago`

`\FN@fnsymbol@wiley`

`\FN@fnsymbol@lamport-robust`

`\FN@fnsymbol@lamport`

```

of Lamport's original
812 \DefineFNsymbols*{bringhurst}{*\dagger\ddagger\S\|P}%
813 \DefineFNsymbols*{chicago}{*\dagger\ddagger\S\|#}%
814 \DefineFNsymbols*{wiley}{*{**}\dagger\ddagger\S\|P\|}%
815 \DefineFNsymbols{lamport-robust}{*\dagger\ddagger
816 \mathsection\mathparagraph\|}%
817 {**}{\dagger\dagger}{\ddagger\ddagger}%
818 }
819 \DefineFNsymbols*{lamport*}{*\dagger\ddagger
820 \mathsection\mathparagraph\|}%
821 {**}{\dagger\dagger}{\ddagger\ddagger}%
822 {\mathsection\mathsection}{\mathparagraph\mathparagraph}%
823 {*{**}}{\dagger\dagger\dagger}{\ddagger\ddagger\ddagger}%
824 {\mathsection\mathsection\mathsection}%
825 {\mathparagraph\mathparagraph\mathparagraph}%
826 }
827 \setfnsymbol{lamport*}
828 \DefineFNsymbols{lamport*-robust}{*\dagger\ddagger
829 \mathsection\mathparagraph\|}%
830 {**}{\dagger\dagger}{\ddagger\ddagger}%
831 {\mathsection\mathsection}{\mathparagraph\mathparagraph}%
832 {*{**}}{\dagger\dagger\dagger}{\ddagger\ddagger\ddagger}%
833 {\mathsection\mathsection\mathsection}%
834 {\mathparagraph\mathparagraph\mathparagraph}%
835 }

```

8 Other miscellaneous commands

8.1 Footnote references

`\footref` Syntax: `\footref{<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`).

`\footref` addresses this problem by making a label reference that actually looks like a `\footnotemark`. (The command is available in the `memoir` class, and we therefore `\providecommand` it rather than defining it “outright”).

```

836 \providecommand*{\footref}[1]{%
837 \begingroup
838 \unrestored@protected@xdef\@thefnmark{\ref{#1}}%
839 \endgroup
840 \@footnotemark
841 }

```

8.2 Minipage `\footnotemarks`

`\mpfootnotemark` Syntax: `\mpfootnotemark[<number>]`

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

```

842 \newcommand\mpfootnotemark[%
843 \@ifnextchar[%

```

```

844 \xmpfootnotemark
845 {%
846   \stepcounter\@mpfn
847   \protected@xdef\@thefnmark{\thempfn}%
848   \@footnotemark
849   }%
850 }
851 \def\xmpfootnotemark[#1]{%
852   \begingroup
853   \csname c@\@mpfn\endcsname #1\relax
854   \unrestored@protected@xdef\@thefnmark{\thempfn}%
855   \endgroup
856   \@footnotemark
857 }

858 \endinput
859 \endpackage

```