

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

Bastien Roucariès[†]formerly Robin Fairbairns

December 12, 2016

Contents

1	User interface — package options	3
1.1	Option <code>perpage</code>	3
1.2	Option <code>para</code>	4
1.3	Option <code>side</code>	4
1.4	Option <code>ragged</code> and <code>\footnotelayout</code>	4
1.5	Option <code>symbol</code>	4
1.6	Option <code>symbol*</code>	5
1.7	The <code>\setfnsymbol</code> and <code>\DefineFNsymbols</code> commands	5
1.8	Option <code>bottom</code>	6
1.9	Option <code>marginal</code>	6
1.10	Option <code>flushmargin</code>	6
1.11	Option <code>hang</code>	6
1.12	Option <code>norule</code>	6
1.13	Option <code>splitrule</code>	7
1.14	The <code>stable</code> option	7
1.15	The <code>multiple</code> option	7
1.16	User interface — miscellaneous commands	8
2	User interface — interactions with other packages	8
3	Code: Preliminaries	9
3.1	Error messages	9
3.2	Check package loaded and package version	9
3.2.1	Helpers	9
3.2.2	Checks	10
3.3	Check definition	10
3.3.1	Helper	10
3.3.2	Checks	12
3.4	Load hook now and at end of some package	13

*This file has version number ?, last revised ?

[†]roucaries.bastien+footmisx@gmail.com

3.5	Disable patching in some package	13
3.6	Generalities	13
4	Package options	14
4.1	The <code>symbol</code> option	14
4.2	The <code>symbol*</code> option	14
4.3	The <code>para</code> option	14
4.4	The <code>side</code> option	15
4.5	The <code>ragged</code> option	15
4.6	The <code>perpage</code> option	15
4.7	The <code>bottom</code> option	16
4.8	The <code>marginal</code> option	16
4.9	The <code>flushmargin</code> option	16
4.10	The <code>hang</code> option	16
4.11	The <code>norule</code> option	16
4.12	The <code>splitrule</code> option	17
4.13	The <code>stable</code> option	17
4.14	The <code>multiple</code> option	18
4.15	The start of the endgame	18
5	Hacking kernel commands	18
5.1	The output routine	18
5.2	Support code for paragraph footnotes	22
5.3	The other footnote commands	24
6	Remaining requirements	25
6.1	The code that executes the <code>multiple</code> option	25
6.2	The code that executes the <code>stable</code> option	26
7	Symbol option variants	27
8	Core function	32
8.1	Tools	32
8.1.1	<code>_@_noHref:n</code>	32
8.2	<code>\@@_makefnmarklink</code>	32
8.3	<code>\@makefnmark</code>	34
8.4	<code>\@footnotemark</code>	34
8.5	<code>\footnotemark</code>	35
8.6	<code>\@footnotetext</code>	35
8.6.1	Hooks	35
8.6.2	<code>setspace</code> package	35
8.6.3	Margin footnote command	36
8.6.4	Paragraph footnote command	36
8.6.5	Normal footnote	37
9	Other miscellaneous commands	38
9.1	Footnote mark style	38
9.2	Footnote references	38
9.3	Minipage <code>\footnotemarks</code>	39

Copyright statement

Program: `footmisx.dtx`

Copyright 2016 Bastien Roucariès Copyright 1995 1996 1998 1999 2001–2003 2008 2009 Robin Fairbairns

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 is based of Robin Fairbairns `footmisc` package.

This package originated as support of a personal project, which Robin was switching to \LaTeX 2 ϵ over the Christmas holiday period of 1993, using the first β 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”.

In 2016 it was forked as `footmisx`. Last version could be found at <https://github.com/bastien-roucaries/footmisx>

1 User interface — package options

The `footmisx` package provides several different customisations of the way footnotes are represented in \LaTeX 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*]{footmisx}`.

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

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 \LaTeX 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” L^AT_EX installations may have problems with the algorithm for `para` footnotes on very wide pages (for example, those used by the `a0poster` class). Recent L^AT_EX 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 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 L^AT_EX would normally set the footnotes a mere `\skip\footins` distant from the bottom of the text.

There's a further infelicity in L^AT_EX's placing of footnotes of the bottom of pages: if a bottom float appears on a page, L^AT_EX 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 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>
<code>\hangfootparindent</code>	<code>0em</code>

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:

<code>\mpfootnoterule</code>	for use in minipages
<code>\pagefootnoterule</code>	for normal footnotes on regular pages
<code>\splitfootnoterule</code>	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, L^AT_EX'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, L^AT_EX 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}
... potato head\footref*{fnlabel} without link.
```

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 way line spacing is calculated in footnotes. `footmisc` knows about this, and preserves the change.

Note that `setspace` could be loaded anywhere except between `hyperref` and `footmisc`.

memoir class The class emulates `setspace`, and we detect that emulation and deal with it in the same way as `setspace`.

hyperref The `hyperref` package has ambitions to make hyperlinks from footnote marks to the corresponding footnote body; `footmisc` knows about this, and preserves hyperlink.

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.

Know incompatibilities are:

footnpag Replace with option `perpage`. You could also use package `perpage` and `\MakePerPagefootnote` command.

pagefoots Replace with option `perpage`. You could also use package `perpage` and `\MakePerPagefootnote` command.

pfnote Replace with option perpage. You could also use package perpage and \MakePerPagefootnote command.

fnpara Replace with option para.

fnpos Replace with option bottom.

3 Code: Preliminaries

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

```
1 <*driver | package>
   Enable lt3docstrip stripping
2 <@@=footmisx>
```

3.1 Error messages

Here we define some error messages:

```
3 \ExplSyntaxOn
4 \msg_new:nnn {footmisx} {not expected definition}
5   {The~macro~#1~is~not~defined~as~expected.\\
6    The~macro~#1~could~not~be~modified~by~footmisx.\\
7    Please~report~a~bug. \\
8    Get:\\
9    #2\\
10   Expect:\\
11   #3\\
12  }
13 \msg_new:nnn {footmisx} {too old}
14   {The~package~#1~is~too~old~and~thus~not~supported.\\
15    Please~install~a~newer~version~(at~least~version~#2)}
16 \msg_new:nnn {footmisx} {incompatible}
17   {The~package~#1~is~incompatible~with~footmisx.\\
18    Use~the~following~workaround~#2}
```

3.2 Check package loaded and package version

3.2.1 Helpers

Protect against too old package by checking version if package is loaded.

```
\___footmisx_error_if_too_old:NN \___footmisx_error_if_too_old:NN {<package>} {<version>}
```

Check minimal version and raise an error in case of too old.

```
19 \cs_new_nopar:Nn \___footmisx_error_if_too_old:NN {
20   \AtBeginDocument{
21     \ExplSyntaxOn
22     \ifpackageloaded{#1}{
23       \ifpackagelater{#1}{#2}{}
24       {\msg_error:nnnn{footmisx}{too old}{#1}{#2}}
25     }{}
26     \ExplSyntaxOff
27   }
28 }
```

Protect against incompatible package.

```
\___footmisx_error_if_incompatible:NN \___footmisx_error_if_incompatible:NN {\package} {\version}
```

Check minimal version and raise an error in case of too old.

```
29 \cs_new_nopar:Nn \___footmisx_error_if_incompatible:NN {
30   \@ifpackageloaded{#1}{
31     \msg_error:nnnn{footmisx}{incompatible}{#1}{#2}
32   }{}
33   \AtBeginOfPackageFile{#1}{
34     \msg_error:nnnn{footmisx}{incompatible}{#1}{#2}
35   }
36   \AtBeginDocument{
37     \ExplSyntaxOn
38     \@ifpackageloaded{#1}{
39       \msg_error:nnnn{footmisx}{incompatible}{#1}{#2}
40     }{}
41     \ExplSyntaxOff
42   }
43 }
```

3.2.2 Checks

Check now some minimal supported version

```
44 \___footmisx_error_if_too_old:NN{hyperref}{2016/06/24}
45 \___footmisx_error_if_too_old:NN{setspace}{2011/12/19}
46 \___footmisx_error_if_incompatible:NN{footnpag}{Use footmisx with option perpage}
47 \___footmisx_error_if_incompatible:NN{pagefoots}{Use footmisx with option perpage}
48 \___footmisx_error_if_incompatible:NN{pfnotes}{Use footmisx with option perpage}
49 \___footmisx_error_if_incompatible:NN{fnpara}{Use footmisx with option para}
50 \___footmisx_error_if_incompatible:NN{fnpos}{Use footmisx with option bottom}
```

3.3 Check definition

3.3.1 Helper

We check definition of some overwritten function. Protect against too recent package.

```
\___footmisxmsg_if_neq:NNc \___footmisxmsg_if_neq:NNc {\sequence_1} {\sequence_2} {\csname}
```

Compare $\{\langle sequence_1 \rangle\}$ and $\{\langle sequence_2 \rangle\}$. Call $\{\langle csname \rangle\}$ if not equal in `\cs_if_eq:NNTF` sense.

$\{\langle csname \rangle\}$ should be a `\foo:nnxxx` function like `\msg_critical:nnxxx`

```
51 \cs_new:Npn \___footmisxmsg_if_neq:NNc #1#2#3 {
52   \cs_if_eq:NNTF {#1} {#2} {}
53     {\use:c{#3}
54      {footmisx}
55      {not expected definition}
56      {\exp_not:N #1}
57      {\exp_not:N {\cs_meaning:N #1}}
58      {\exp_not:N {\cs_meaning:N #2}}}
59   }
60 }
```

```
\___footmisx_fatal_if_neq:NN \___footmisx_fatal_if_neq:NN {\sequence_1} {\sequence_2}
```

Compare $\{\langle sequence_1 \rangle\}$ and $\{\langle sequence_2 \rangle\}$. Call `\msg_fatal:nnxxx` if not equal in `\cs_if_eq:NNTF` sense.

```
61 \cs_new:Nn \___footmisx_fatal_if_neq:NN {
62   \___footmisx_msg_if_neq:NNc {#1} {#2} {msg_fatal:nnxxx}
63 }
```

```
\___footmisx_critical_if_neq:NN \___footmisx_critical_if_neq:NN {\sequence_1} {\sequence_2}
```

Compare $\{\langle sequence_1 \rangle\}$ and $\{\langle sequence_2 \rangle\}$. Call `\msg_critical:nnxxx` if not equal in `\cs_if_eq:NNTF` sense.

```
64 \cs_new:Nn \___footmisx_critical_if_neq:NN {
65   \___footmisx_msg_if_neq:NNc {#1} {#2} {msg_critical:nnxxx}
66 }
```

```
\___footmisxerror_if_neq:NN \___footmisxerror_if_neq:NN {\sequence_1} {\sequence_2}
```

Compare $\{\langle sequence_1 \rangle\}$ and $\{\langle sequence_2 \rangle\}$. Call `\msg_error:nnxxx` if not equal in `\cs_if_eq:NNTF` sense.

```
67 \cs_new:Nn \___footmisxerror_if_neq:NN {
68   \___footmisx_msg_if_neq:NNc {#1} {#2} {msg_error:nnxxx}
69 }
```

```
\___footmisx_warning_if_neq:NN \___footmisx_warning_if_neq:NN {\sequence_1} {\sequence_2}
```

Compare $\{\langle sequence_1 \rangle\}$ and $\{\langle sequence_2 \rangle\}$. Call `\msg_warning:nnxxx` if not equal in `\cs_if_eq:NNTF` sense.

```
70 \cs_new:Nn \___footmisx_warning_if_neq:NN {
71   \___footmisx_msg_if_neq:NNc {#1} {#2} {msg_warning:nnxxx}
72 }
```

```
\_footmisx_note_if_neq:NN \_footmisx_note_if_neq:NN {\sequence_1} {\sequence_2}
```

Compare $\{\langle sequence_1 \rangle\}$ and $\{\langle sequence_2 \rangle\}$. Call `\msg_note:nnxxx` if not equal in `\cs_if_eq:NNTF` sense.

```
73 \cs_new:Nn \_footmisx_note_if_neq:NN {
74   \_footmisx_msg_if_neq:NNc {#1} {#2} {msg_note:nnxxx}
75 }
```

```
\___footmisx_log_if_neq:NN \___footmisx_log_if_neq:NN {\sequence_1} {\sequence_2}
```

Compare $\{\langle sequence_1 \rangle\}$ and $\{\langle sequence_2 \rangle\}$. Call `\msg_log:nnxxx` if not equal in `\cs_if_eq:NNTF` sense.

```
76 \cs_new:Nn \___footmisx_log_if_neq:NN {
77   \___footmisx_msg_if_neq:NNc {#1} {#2} {msg_log:nnxxx}
78 }
```

3.3.2 Checks

Now we check if `\@makefnmark` has been modified. Do it between group in order to not pollute namespace.

```

79 \group_begin:
80 \def\__footmisx_copy_@makefnmark{\hbox{\@textsuperscript{\normalfont\@thefnmark}}}%
81 \__footmisx_warning_if_neq:NN {\@makefnmark} {\__footmisx_copy_@makefnmark}
82 \group_end:

```

Now we check if `\@footnotemark` has been modified. Do it between group in order to not pollute namespace.

```

83 \group_begin:
84 \@ifpackageloaded{hyperref}%
85 %%hyperref case
86 {
87   \def\__footmisx_copy_@footnotemark{%
88     \leavevmode
89     \ifhmode\edef\@x@sf{\the\spacefactor}\nobreak\fi
90     \stepcounter{Hfootnote}%
91     \global\let\Hy@saved@currentHref\@currentHref
92     \hyper@makecurrent{Hfootnote}%
93     \global\let\Hy@footnote@currentHref\@currentHref
94     \global\let\@currentHref\Hy@saved@currentHref
95     \hyper@linkstart{link}{\Hy@footnote@currentHref}%
96     \@makefnmark
97     \hyper@linkend
98     \ifhmode\spacefactor\@x@sf\fi
99     \relax
100   }%
101 }
102 %% latex case
103 {
104   \def\__footmisx_copy_@footnotemark{
105     \leavevmode
106     \ifhmode\edef\@x@sf{\the\spacefactor}\nobreak\fi
107     \@makefnmark
108     \ifhmode\spacefactor\@x@sf\fi
109     \relax
110   }
111 }
112 \__footmisxerror_if_neq:NN {\@footnotemark} {\__footmisx_copy_@footnotemark}
113 \group_end:

```

Now we check if `\footnotemark` has been modified. Do it between group in order to not pollute namespace.

```

114 \group_begin:
115 \def\__footmisx_copy_footnotemark{\@ifnextchar [%]
116   \@xfootnotemark
117   {%
118     \stepcounter{footnote}%
119     \protected@xdef\@thefnmark{\thefootnote}%
120     \@footnotemark
121   }%
122 }

```

```

123 \___footmisxerror_if_neq:NN {\footnotemark} {\___footmisx_copy_footnotemark}
124 \group_end:

```

3.4 Load hook now and at end of some package

```

\___footmisx_AtEndOfPackageFile_and_now:NN \___footmisx_AtEndOfPackageFile_and_now:NN {\package}
{\code}

```

Run `{\code}` now and at end of package `{\package}`.

```

125 \cs_new:Npn \___footmisx_AtEndOfPackageFile_and_now:NN #1#2 {
126 #2
127 \AtEndOfPackageFile{#1}{#2}
128 }

```

3.5 Disable patching in some package

```

\___footmisx_disablepatchinginpackage:cN \___footmisx_disablepatchinginpackage:cN {\function}
{\package}

```

Save function at beginning of package and restore at end. Create two global token list:

- `___footmisx_save_function@package_tl` that save `{\function}` to `\@_saved_function@package_tl`.
- `___footmisx_restore_function@package_tl` that restore `{\function}` from `\@_saved_function@package_tl`.

```

129 \cs_new:Npn \___footmisx_disablepatchinginpackage:cN #1#2 {
130   \tl_const:cn {\___footmisx_save_ #1 @ #2 _tl}{
131     \cs_gset_eq:cc {\___footmisx_saved_ #1 @ #2 _tl} {#1}
132   }
133   \tl_const:cn {\___footmisx_restore_ #1 @ #2 _tl}{
134     \cs_gset_eq:cc {#1} {\___footmisx_saved_ #1 @ #2 _tl}
135   }
136   \AtBeginOfPackageFile{#2}{\use:c{\___footmisx_save_ #1 @ #2 _tl}}
137   \AtEndOfPackageFile{#2}{\use:c{\___footmisx_restore_ #1 @ #2 _tl}}
138 }

```

3.6 Generalities

We save a private macro using `csname` in order to use `lt3doc`

```

139 \cs_set_eq:cc {\___footmisx_par} {@ @ par}
140 \ExplSyntaxOff

```

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

```

141 \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).

```

142 \providecommand\protected@writeaux{%
143   \protected@write\@auxout
144 }

```

(End definition for `\protected@writeaux`. This function is documented on page ??.)

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

```
145 \newdimen\footnotemargin
146 \footnotemargin1.8em\relax
```

(End definition for `\footnotemargin`. This function is documented on page ??.)

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.

```
147 \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).

```
148 \DeclareOption{symbol*}{%
149   \renewcommand\thefootnote{\@fnsymbol\c@footnote}%
150   \AtEndOfPackage{\setfnsymbol{lamport*-robust}}}%
151 }
```

4.3 The `para` option

The basis of the code for this option comes from T_EXbook, p.398 ff. (“Dirty Tricks”), though it does (of course) avoid redefining `\` which has some other (somewhat significant) uses in L^AT_EX! The user should be aware of Knuth’s note on the limitations of this method of doing the job: the T_EX stack is used four times per footnote, and the stack is limited (see the T_EXbook, 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 L^AT_EX, 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.

```
152 \newif\ifFN@para \FN@parafalse
153 \DeclareOption{para}{\ifFN@sidefn
154   \PackageError{footmisx}{Option "CurrentOption" incompatible with
155     option "side"}%
```

```

156     {I shall ignore "\CurrentOption"}%
157   \else
158     \FN@paratrue
159   \fi
160 }

```

(End definition for \ifFN@para. This function is documented on page ??.)

4.4 The side option

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

```

161 \newif\ifFN@sidefn \FN@sidefnfalse
162 \DeclareOption{side}{\ifFN@para
163   \PackageError{footmisx}{Option "\CurrentOption" incompatible with
164     option "para"}%
165   {I shall ignore "\CurrentOption"}%
166 \else
167   \FN@sidefntrue
168 \fi
169 }

```

(End definition for \ifFN@sidefn. This function is documented on page ??.)

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.

```

170 \let\footnotelayout\@empty
171 \DeclareOption{ragged}{%
172   \@ifundefined{RaggedRight}%
173   {\renewcommand\footnotelayout{\linepenalty50 \raggedright}}%
174   {\renewcommand\footnotelayout{\linepenalty50 \RaggedRight}}%
175 }

```

(End definition for \footnotelayout. This function is documented on page ??.)

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 5.3).

```

176 \newif\ifFN@perpage
177 \FN@perpagefalse
178 \DeclareOption{perpage}{%
179   \FN@perpagetrue
180 }

```

(End definition for \ifFN@perpage. This function is documented on page ??.)

4.7 The bottom option

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

```
181 \newif\ifFN@bottom \FN@bottomfalse
182 \DeclareOption{bottom}{%
183   \FN@bottomtrue
184 }
```

(End definition for \ifFN@bottom. This function is documented on page ??.)

4.8 The marginal option

Again, the processing of the option is pretty trivial:

```
185 \DeclareOption{marginal}{%
186   \footnotemargin-0.8em\relax
187 }
```

4.9 The flushmargin option

Again, the processing of the option is pretty trivial:

```
188 \DeclareOption{flushmargin}{%
189   \footnotemargin0pt\relax
190 }
```

4.10 The hang option

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

```
191 \newif\ifFN@hangfoot \FN@hangfootfalse
192 \DeclareOption{hang}{%
193   \FN@hangfoottrue
194 }
```

(End definition for \ifFN@hangfoot. This function is documented on page ??.)

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

```
195 \newcommand*\hangfootparskip{0.5\baselineskip}
196 \newcommand*\hangfootparindent{0em}%
```

(End definition for \hangfootparskip and \hangfootparindent. These functions are documented on page ??.)

4.11 The norule option

Pretty simple too...

```
197 \DeclareOption{norule}{%
198   \renewcommand\footnoterule{}%
199   \advance\skip\footins 4\p@\@plus2\p@\relax
200 }
```


4.12 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.

```
201 \DeclareOption{splitrule}{%
202   \gdef\split@prev{0}}
```

(End definition for \split@prev. This function is documented on page ??.)

`\pagefootnoterule` Define defaults for the three footnote rules: note, we inherit the current state of
`\mpfootnoterule` `\footnoterule` for the two ‘regular’ footnote defaults, and if we’ve been preceded by
`\splitfootnoterule` option `norule`, they will both become null...

```
203 \let\pagefootnoterule\footnoterule
204 \let\mpfootnoterule\footnoterule
205 \def\splitfootnoterule{\kern-3\p@ \hrule \kern2.6\p@}
```

(End definition for \pagefootnoterule, \mpfootnoterule, and \splitfootnoterule. These functions are documented on page ??.)

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

```
206 \def\footnoterule{\relax
207   \ifx \@listdepth\@mplistdepth
```

In a minipage

```
208     \mpfootnoterule
209   \else
210     \ifnum\split@prev=\z@
```

Normal footnote on a regular page

```
211     \pagefootnoterule
212   \else
```

Second part of a split footnote

```
213     \splitfootnoterule
214   \fi
```

Remember a split for next page

```
215     \xdef\split@prev{\the\insertpenalties}%
216   \fi
217 }%
218 }
```

`\ifFN@stablefootnote` 4.13 The stable option

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

```
219 \newif\ifFN@stablefootnote \FN@stablefootnotefalse
220 \DeclareOption{stable}{\FN@stablefootnotetrue}
```

(End definition for \ifFN@stablefootnote. This function is documented on page ??.)

4.14 The multiple option

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

```
221 \newif\ifFN@multiplefootnote \FN@multiplefootnotefalse
222 \DeclareOption{multiple}{\FN@multiplefootnotetrue}
```

(End definition for \ifFN@multiplefootnote. This function is documented on page ??.)

4.15 The start of the endgame

Exercise the options that the user has requested...

```
223 \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.

```
224 \let \if@tempswa \ifFN@bottom
225 \ifFN@para \@tempwattrue \fi
226 \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.)

```
227 \@ifl@t@r\fmtversion{2005/12/01}{%
228   \CheckCommand*\@makecol{\ifvoid \footins
229     \setbox\@outputbox \box\@cclv
230   }%
231   \setbox\@outputbox \vbox{%
232     \boxmaxdepth \@maxdepth
233     \unvbox\@cclv
234     \vskip \skip\footins
235     \color@begingroup
236       \normalcolor\footnoterule
237     \unvbox\footins
238     \color@endgroup
239   }%
240 \fi
241 \let \@elt \relax
242 \xdef\@freelist{\@freelist\@midlist}%
243 \global\let\@midlist\@empty
244 \@combinefloats
245 \ifvbox\@kludgeins
```

```

246         \@makespecialcolbox
247     \else
248         \setbox\@outputbox \vbox to\@colht{%
249             \@texttop \dimen@\dp\@outputbox
250             \unvbox\@outputbox
251             \vskip -\dimen@\@textbottom
252         }%
253     \fi
254     \global\maxdepth\@maxdepth
255 }
256 }{%
257 \@ifl@t@r\fmtversion{2003/12/01}{%
258     \CheckCommand*\@makecol{\ifvoid \footins
259         \setbox\@outputbox \box\@ccclv
260     \else
261         \setbox\@outputbox \vbox{%
262             \boxmaxdepth\@maxdepth
263             \@tempdima\dp\@ccclv
264             \unvbox\@ccclv
265             \vskip \skip\footins
266             \color@begingroup
267             \normalcolor
268             \footnoterule
269             \unvbox\footins
270             \color@endgroup
271         }%
272     \fi
273     \let \@elt \relax
274     \xdef\@freelist{\@freelist\@midlist}%
275     \global\let\@midlist\@empty
276     \@combinefloats
277     \ifvbox\@kludgeins
278         \@makespecialcolbox
279     \else
280         \setbox\@outputbox \vbox to\@colht{%
281             \@texttop
282             \dimen@\dp\@outputbox
283             \unvbox\@outputbox
284             \vskip -\dimen@
285             \@textbottom
286         }%
287     \fi
288     \global\maxdepth\@maxdepth
289 }%
290 }{%
291 \@ifl@t@r\fmtversion{1999/12/01}{%
292     \CheckCommand*\@makecol{\ifvoid \footins
293         \setbox\@outputbox \box\@ccclv
294     \else
295         \setbox\@outputbox \vbox{%
296             \boxmaxdepth\@maxdepth
297             \@tempdima\dp\@ccclv
298             \unvbox\@ccclv
299             \vskip \skip\footins

```

```

300         \color@begingroup
301         \normalcolor\footnoterule
302         \unvbox\footins
303         \color@endgroup
304     }%
305 \fi
306 \xdef\@freelist{\@freelist\@midlist}%
307 \global\let\@midlist\@empty
308 \@combinefloats
309 \ifvbox\@kludgeins
310     \@makespecialcolbox
311 \else
312     \setbox\@outputbox \vbox to\@colht{%
313         \@texttop \dimen@\dp\@outputbox
314         \unvbox\@outputbox
315         \vskip -\dimen@\@textbottom
316     }%
317 \fi
318 \global\maxdepth\@maxdepth
319 }%
320 }%
321 }{%
322     \CheckCommand*\@makecol{\ifvoid \footins
323         \setbox\@outputbox \box\@cclv
324     \else
325         \setbox\@outputbox \vbox{%
326             \boxmaxdepth\@maxdepth
327             \unvbox\@cclv
328             \vskip \skip\footins
329             \color@begingroup
330             \normalcolor\footnoterule
331             \unvbox\footins
332             \color@endgroup
333         }%
334     \fi
335     \xdef\@freelist{\@freelist\@midlist}%
336     \global\let\@midlist\@empty
337     \@combinefloats
338     \ifvbox\@kludgeins
339         \@makespecialcolbox
340     \else
341         \setbox\@outputbox \vbox to\@colht{%
342             \@texttop \dimen@\dp\@outputbox
343             \unvbox\@outputbox
344             \vskip -\dimen@\@textbottom
345         }%
346     \fi
347     \global\maxdepth\@maxdepth
348 }%
349 }%
350 }

```

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

```
351 \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`.

```
352 \FN@temptoken{%
353   \toks@\expandafter{\the\toks@
354     \vskip\skip\footins
355     \color@begingroup
356     \normalcolor\footnoterule
357     \global\setbox\FN@tempboxc\vbox{\makefootnoteparagraph}%
358     \unvbox\FN@tempboxc
359   \color@endgroup
360 }%
361 }%
```

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

```
362 \else
363   \FN@temptoken{%
364     \toks@\expandafter{\the\toks@
365       \vskip\skip\footins
366       \color@begingroup
367       \normalcolor\footnoterule
368       \unvbox\footins
369     \color@endgroup
370   }%
371   }%
372 \fi
```

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

```
373 \ifFN@bottom
374   \toks@{\setbox\@outputbox \box\@cclv
375     \xdef\@freelist{\@freelist\@midlist}%
376     \global\let\@midlist\@empty
377     \@combinefloats
378     \ifvoid\footins
379     \else
380       \setbox\@outputbox \vbox\bgroup
381         \boxmaxdepth\@maxdepth
382         \unvbox\@outputbox
383         \vfill\relax
384     }
385   \the\FN@temptoken
386   \toks@\expandafter{\the\toks@\egroup\fi}
```

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

```
387 \else
388   \toks@{\ifvoid\footins
389     \setbox\@outputbox\box\@cclv
390   \else
391     \setbox\@outputbox \vbox\bgroup
```

```

392         \boxmaxdepth\@maxdepth
393         \unvbox\@cclv
394     }
395     \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@`.

```

396     \toks@\expandafter{\the\toks@
397         \egroup
398         \fi
399         \xdef\@freelist{\@freelist\@midlist}%
400         \global\let\@midlist\@empty
401         \@combinefloats
402     }%
403     \fi

```

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

```

404     \toks@\expandafter{\the\toks@
405         \ifvbox\@kludgeins
406             \@makespecialcolbox
407         \else
408             \setbox\@outputbox \vbox to\@colht{%
409                 \@texttop \dimen@dp\@outputbox
410                 \unvbox\@outputbox
411                 \vskip -\dimen@\@textbottom
412             }%
413         \fi
414         \global\maxdepth\@maxdepth
415     }
416     \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.

```

417 \fi

```

5.2 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.

```

418 \ifFN@para

```

`\FN@tempboxa` We need some temporary boxes, and L^AT_EX only defines one

```

\FN@tempboxb 419 \let\FN@tempboxa\FN@tempboxa
\FN@tempboxb 420 \newbox\FN@tempboxb
421 \newbox\FN@tempboxc

```

(End definition for `\FN@tempboxa`, `\FN@tempboxb`, and `\FN@tempboxc`. These functions are documented on page ??.)

`\footglue` A direct crib from the T_EXbook:

```

422 \newskip\footglue \footglue=1em plus.3em minus.3em

```

(End definition for `\footglue`. This function is documented on page ??.)

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

```
423 \long\def\@makefntext#1{\leavevmode
424   \textfootmark{\@thefnmark}\nobreak
425   \hskip.5em\relax#1%
426 }
```

(End definition for \@makefntext. This function is documented on page ??.)

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

```
427 \newdimen\footnotebaselineskip
428 {%
429   \footnotesize
430   \global
431   \footnotebaselineskip=\normalbaselineskip
432 }
```

(End definition for \footnotebaselineskip. This function is documented on page ??.)

`\makefootnoteparagraph` For use in the output routine

```
433 \long\def\makefootnoteparagraph{\unvbox\footins \makehboxoffhboxes
434   \setbox\FN@tempboxa=\hbox{\unhbox\FN@tempboxa \removehboxes}
```

Now we are ready to set the paragraph:

```
435   \hsize\columnwidth
436   \@parboxrestore
437   \baselineskip=\footnotebaselineskip
438   \noindent
439   \rule{\z@}{\footnotesep}%
440   \unhbox\FN@tempboxa\par
441 }
```

(End definition for \makefootnoteparagraph. This function is documented on page ??.)

`\makehboxoffhboxes` Support code for `\makefootnoteparagraph`

`\removehboxes`

```
442 \def\makehboxoffhboxes{\setbox\FN@tempboxa=\hbox{}}%
443 \loop
444   \setbox\FN@tempboxb=\lastbox
445   \ifhbox\FN@tempboxb
446     \setbox\FN@tempboxa=\hbox{\box\FN@tempboxb\unhbox\FN@tempboxa}%
447   \repeat
448 }
449 \def\removehboxes{\setbox\FN@tempboxa=\lastbox
450   \ifhbox
451     \FN@tempboxa{\removehboxes}%
452   \unhbox\FN@tempboxa
453   \fi
454 }
455 \fi
```

(End definition for \makehboxoffhboxes and \removehboxes. These functions are documented on page ??.)

5.3 The other footnote commands

```

456 \iffN@perpage
457   \RequirePackage{perpage}
458   \MakePerPage{footnote}
459 \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.

```

460 \iffN@para
461 \else
    hanging footnote version:
462   \long\def\@makefnstext#1{%
463     \iffN@hangfoot
464     \bgroup
    get the marker so we can measure it:
465     \setbox\@tempboxa\hbox{%
466       \ifdim\footnotemargin>0pt
467         \hb@xt@\footnotemargin{\textfootmark{\@thefnmark}\hss}%
468       \else
469         \textfootmark{\@thefnmark}
470       \fi
471     }%

```

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(!)`

```

472     \leftmargin\wd\@tempboxa
473     \rightmargin\z@
474     \linewidth \columnwidth
475     \advance \linewidth -\leftmargin
476     \parshape \@ne \leftmargin \linewidth
477     \footnotesize

```

stop the `\parshape` being overwritten:

```

478     \@setpar{\_footmisxpar}}%

```

and finally put the marker in its chosen place:

```

479     \leavevmode
480     \llap{\box\@tempboxa}%
481     \parskip\hangfootparskip\relax
482     \parindent\hangfootparindent\relax
483   \else

```

ordinary (non-hanging) footnote version:

```

484     \parindent1em
485     \noindent
486     \ifdim\footnotemargin>\z@
487       \hb@xt@ \footnotemargin{\hss\textfootmark{\@thefnmark}}%
488     \else
489       \ifdim\footnotemargin=\z@
490         \llap{\textfootmark{\@thefnmark}}%
491       \else
492         \llap{\hb@xt@ -\footnotemargin{\textfootmark{\@thefnmark}\hss}}%
493       \fi

```



```

494     \fi
495     \fi
496     \footnotelayout#1%
    if we're hanging, close the hang group
497     \ifFN@hangfoot
498     \par\egroup
499     \fi
500 }
501 \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

<pre> \multiplefootnotemarker \multfootsep \@footnotemark \FN@mf@prepare \FN@mf@check </pre>	<p>This (revised) code derives from a suggestion by Alexander Rozhenko (the author of the <i>manyfoot</i> package): the intention is that <i>footmisc</i> and <i>manyfoot</i> 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 <code>\footnote</code> and <code>\footnotemark</code> insert a marker (a cancelling pair of kerns of <code>\multiplefootnotemarker</code> (of opposite signs), which is detected in following <code>\footnote</code> or <code>\footnotemark</code> 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.</p>
--	--

```

502 \ifFN@multiplefootnote
503   \providecommand*{\multiplefootnotemarker}{3sp}
504   \providecommand*{\multfootsep}{,}
505   \def\FN@mf@prepare{%
506     \kern-\multiplefootnotemarker
507     \kern\multiplefootnotemarker\relax
508   }
509   \def\FN@mf@check{%
510     \ifdim\lastkern=\multiplefootnotemarker\relax
511       \edef\x@sf{\the\spacefactor}%
512       \unkern
513       \textsuperscript{\multfootsep}%
514       \spacefactor\x@sf\relax
515     \fi
516   }

```

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

```

517 \else
518   \let\FN@mf@check\relax
519   \let\FN@mf@prepare\relax
520 \fi

```

(End definition for `\multiplefootnotemarker` and others. These functions are documented on page ??.)

6.2 The code that executes the stable option

`\ifFN@stablefootnote`
`\FN@sfootmisx@footnote` 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’.

```

521 \ifFN@stablefootnote
522 \let\FN@sfootmisx@footnote\footnote
523 \def\footnote{\ifx\protect\@typeset@protect
524   \expandafter\FN@sfootmisx@footnote
525   \else
526   \expandafter\FN@sfootgobble@opt
527   \fi
528 }
```

(End definition for \ifFN@stablefootnote and \FN@sfootmisx@footnote. These functions are documented on page ??.)

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

```

529 \edef\FN@sfootgobble@opt{\noexpand\protect
530   \expandafter\noexpand\csname FN@sfootgobble@opt \endcsname}
531 \expandafter\def\csname FN@sfootgobble@opt \endcsname{%
532   \@ifnextchar[%]
533   \FN@sfootgobble@twobracket
534   \gobble
535 }
536 \def\FN@sfootgobble@twobracket[#1]#2{}
```

(End definition for \FN@sfootgobble@opt and \FN@sfootgobble@twobracket. These functions are documented on page ??.)

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

```

\FN@sfootgobble@optonly 537 \let\FN@sfootmisx@footnotemark\footnotemark
\FN@sfootgobble@bracket 538 \def\footnotemark{\ifx\protect\@typeset@protect
539   \expandafter\FN@sfootmisx@footnotemark
540   \else
541   \expandafter\FN@sfootgobble@optonly
542   \fi
543 }
544 \edef\FN@sfootgobble@optonly{\noexpand\protect
545   \expandafter\noexpand\csname FN@sfootgobble@optonly \endcsname}
546 \expandafter\def\csname FN@sfootgobble@optonly \endcsname{%
547   \@ifnextchar[%]
548   \FN@sfootgobble@bracket
549   {}%
550 }
551 \def\FN@sfootgobble@bracket[#1]{ }
552 \fi
```

(End definition for \FN@sfootmisx@footnotemark, \FN@sfootgobble@optonly, and \FN@sfootgobble@bracket. These functions are documented on page ??.)

```
\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.

```
553 \newcommand\setfnsymbol[1]{%
554   \bsphack
555   \ifundefined{FN@fnsymbol@#1}%
556   {%
557     \PackageError{footmisc}{Symbol style "#1" not known}%
558     \@eha
559   }%
560   \expandafter\let\expandafter\FN@fnsymbol\csname
561     FN@fnsymbol@#1\endcsname
562  }%
563   \@esphack
564 }
```

The default selection is Lamport’s original, as represented in current L^AT_EX — we preserve it in case we need to “get back” to it.

```
565 \let\FN@fnsymbol@lamport\FN@fnsymbol
566 \</driver | package>
```

(End definition for \setfnsymbol and \FN@fnsymbol@lamport. These functions are documented on page ??.)

```
\if@tempswb
\@tempswbfalse
\@tempswbtrue
```

We need another temp conditional

```
567 \newif\if@tempswb
```

(End definition for \if@tempswb, \@tempswbfalse, and \@tempswbtrue. These functions are documented on page ??.)

```
\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 L^AT_EX `\fnsymbol` may be expressed this way in a sufficiently old L^AT_EX 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\dagger}{\ddagger\ddagger}%
```

Note that doubled-up (and worse — see below) symbols need braces around them.

```
568 \newcommand{\DefineFNSymbols}{%
569   \ifstar{\@tempswbtrue\@DefineFNSymbols}%
570   {\@tempswbfalse\@DefineFNSymbols}%
```

```

571 }
572 \newcommand{\@DefineFNSymbols}[1]{%
573   \@ifnextchar[% ]
574     {\@DefineFNSymbols@{#1}}{\@DefineFNSymbols@{#1}[text]}}%
575 }
576 \def\@DefineFNSymbols@#1[#2]#3{%
577   \expandafter\ifx\csname FN@fnsymbol@#1\endcsname\relax
578     \PackageInfo{footmisc}{Declaring symbol style #1}%
579   \else
580     \PackageWarning{footmisc}{Redeclaring symbol style #1}%
581   \fi
582   \toks@{}%
583   \def\@tempb{\end}%
584   \FN@build@symboldef#3\end
585   \def\@tempc{math}%
586   \def\@tempd{#2}%
587   \expandafter\xdef\csname FN@fnsymbol@#1\endcsname##1{%
588     \ifx\@tempc\@tempd
589       \noexpand\ensuremath
590     \else
591       \noexpand\nfss@text
592     \fi
593     {%
594       \noexpand\ifcase##1%
595       \the\toks@
596       \noexpand\else
597       \if@tempswb
598         \noexpand\@ctrerr
599       \else
600         \noexpand\@arabic##1\noexpand\FN@orange##1%
601       \fi
602       \noexpand\fi
603     }%
604   }%
605 }
606 \def\FN@build@symboldef#1{%
607   \def\@tempa{#1}%
608   \ifx\@tempa\@tempb
609   \else
610     \toks@\expandafter{\the\toks@\or#1}%
611     \expandafter\FN@build@symboldef
612   \fi
613 }

```

(End definition for `\DefineFNSymbols` and others. These functions are documented on page ??.)

`\DefineFNSymbolsTM` Now do the same job for the “modern” way of having both text and maths variants of everything.

```

\@DefineFNSymbolsTM
\FN@build@symboldefTM
614 \newcommand{\DefineFNSymbolsTM}{%
615   \@ifstar{\@tempswbtrue\@DefineFNSymbolsTM}%
616     {\@tempswbfalse\@DefineFNSymbolsTM}}%
617 \newcommand{\@DefineFNSymbolsTM}[2]{%
618   \expandafter\ifx\csname FN@fnsymbol@#1\endcsname\relax
619     \PackageInfo{footmisc}{Declaring symbol style #1}%

```

```

620 \else
621   \PackageWarning{footmisx}{Redeclaring symbol style #1}%
622 \fi
623 \toks@{}%
624 \def\@tempb{\end}%
625 \FN@build@symboldefTM#2\end\@null
626 \expandafter\xdef\csname FN@fnsymbol@#1\endcsname##1{%
627   \noexpand\ifcase##1%
628     \the\toks@
629   \noexpand\else
630     \if@tempwb
631       \noexpand\@ctrerr
632     \else
633       \noexpand\@arabic##1\noexpand\FN@orange##1%
634     \fi
635   \noexpand\fi
636 }%
637 }%

```

Note that this version has two variants of every definition, so needs two stopper codes above.

```

638 \def\FN@build@symboldefTM#1#2{%
639   \def\@tempa{#1}%
640   \ifx\@tempa\@tempb
641   \else
642     \toks@\expandafter{\the\toks@\or\TextOrMath{#1}{#2}}%
643     \expandafter\FN@build@symboldefTM
644   \fi
645 }

```

(End definition for \DefineFNsymbolsTM, \@DefineFNsymbolsTM, and \FN@build@symboldefTM. These functions are documented on page ??.)

\TextOrMath This is a stripped down (e-TeX only) version of what appears in fixltx2e. If the command's already defined, we assume it's that version.

```

646 \@ifundefined{TextOrMath}{%
647   \@ifundefined{eTeXversion}{%
648     \PackageError{footmisx}{Can't define commands for footnote symbol}%
649     {Use e-LaTeX, or load package fixltx2e before
650      footmisx}%
651   }%
652   \protected\expandafter\def\csname TextOrMath\space\endcsname{%
653     \ifmmode \expandafter\@secondoftwo
654     \else \expandafter\@firstoftwo \fi
655   }
656   \edef\TextOrMath#1#2{%
657     \expandafter\noexpand\csname TextOrMath\space\endcsname
658     {#1}{#2}%
659   }%
660 }%
661 }{}

```

(End definition for \TextOrMath. This function is documented on page ??.)

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

```

662 \def\FN@orange#1{%
663   \@bsphack
664   \PackageInfo{footmisc}{Footnote number \number#1 out of range}%
665   \protect\@fnsymbol@orange
666   \@esphack
667 }
668 \global\let\@diagnose@fnsymbol@orange\relax
669 \AtEndDocument{\@diagnose@fnsymbol@orange}
670 \def\@fnsymbol@orange{%
671   \gdef\@diagnose@fnsymbol@orange{%
672     \PackageWarningNoLine{footmisc}{Some footnote number(s)
673       were out of range
674       \MessageBreak
675       see log for details%
676     }%
677   }%
678 }

```

(End definition for \FN@orange, \@fnsymbol@orange, and \@diagnose@fnsymbol@orange. These functions are documented on page ??.)

`\textbardbl` This is defined in recent L^AT_EX releases, but not in (for example) that distributed with the last release of t_EX. Since it's needed in some symbol set definitions (including Lamport's) we define it here.

```

679 \ifundefined{textbardbl}{%
680   \DeclareTextSymbol{\textbardbl}{OMS}{107}%
681   \DeclareTextSymbolDefault{\textbardbl}{TS1}}{}%

```

(This definition comes from the L^AT_EX sources.)

(End definition for \textbardbl. This function is documented on page ??.)

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

```

682 \DefineFNSymbolsTM*{bringhurst}{%
683   \textasteriskcentered *
684   \textdagger \dagger
685   \textdaggerdbl \ddagger
686   \textsection \mathsection
687   \textbardbl \||%
688   \textparagraph \mathparagraph
689 }%
690 \DefineFNSymbolsTM*{chicago}{%
691   \textasteriskcentered *
692   \textdagger \dagger
693   \textdaggerdbl \ddagger
694   \textsection \mathsection
695   \textbardbl \||%
696   \#\#%
697 }%
698 \DefineFNSymbolsTM*{wiley}{
699   \textasteriskcentered *

```

```

700 {\textasteriskcentered\textasteriskcentered}{**}%
701 \textdagger \dagger
702 \textdaggerdbl \ddagger
703 \textsection \mathsection
704 \textparagraph \mathparagraph
705 \textbardbl \||%
706 }%
707 \DefineFNsymbolsTM{\lamport-robust}{
708 \textasteriskcentered *
709 \textdagger \dagger
710 \textdaggerdbl \ddagger
711 \textsection \mathsection
712 \textparagraph \mathparagraph
713 \textbardbl \||%
714 {\textasteriskcentered\textasteriskcentered}{**}%
715 {\textdagger\textdagger}{\dagger\dagger}%
716 {\textdaggerdbl\textdaggerdbl}{\ddagger\ddagger}%
717 }
718 \DefineFNsymbolsTM*{\lamport*}{%
719 \textasteriskcentered *
720 \textdagger \dagger
721 \textdaggerdbl \ddagger
722 \textsection \mathsection
723 \textparagraph \mathparagraph
724 \textbardbl \||%
725 {\textasteriskcentered\textasteriskcentered}{**}%
726 {\textdagger\textdagger}{\dagger\dagger}%
727 {\textdaggerdbl\textdaggerdbl}{\ddagger\ddagger}%
728 {\textsection\textsection}{\mathsection\mathsection}%
729 {\textparagraph\textparagraph}{\mathparagraph\mathparagraph}%
730 {\textasteriskcentered\textasteriskcentered\textasteriskcentered}{***}%
731 {\textdagger\textdagger\textdagger}{\dagger\dagger\dagger}%
732 {\textdaggerdbl\textdaggerdbl\textdaggerdbl}{\ddagger\ddagger\ddagger}%
733 {\textsection\textsection\textsection}%%
734 {\mathsection\mathsection\mathsection}%
735 {\textparagraph\textparagraph\textparagraph}%%
736 {\mathparagraph\mathparagraph\mathparagraph}%
737 }
738 \setfnsymbol{\lamport*}
739 \DefineFNsymbolsTM{\lamport*-robust}{%
740 \textasteriskcentered *
741 \textdagger \dagger
742 \textdaggerdbl \ddagger
743 \textsection \mathsection
744 \textparagraph \mathparagraph
745 \textbardbl \||%
746 {\textasteriskcentered\textasteriskcentered}{**}%
747 {\textdagger\textdagger}{\dagger\dagger}%
748 {\textdaggerdbl\textdaggerdbl}{\ddagger\ddagger}%
749 {\textsection\textsection}{\mathsection\mathsection}%
750 {\textparagraph\textparagraph}{\mathparagraph\mathparagraph}%
751 {\textasteriskcentered\textasteriskcentered\textasteriskcentered}{***}%
752 {\textdagger\textdagger\textdagger}{\dagger\dagger\dagger}%
753 {\textdaggerdbl\textdaggerdbl\textdaggerdbl}{\ddagger\ddagger\ddagger}%

```

```

754 {\textsection\textsection\textsection}%%
755 {\mathsection\mathsection\mathsection}%
756 {\textparagraph\textparagraph\textparagraph}%%
757 {\mathparagraph\mathparagraph\mathparagraph}%
758 }

```

(End definition for \FN@fnsymbol@bringhurst and others. These functions are documented on page ??.)

8 Core function

8.1 Tools

Here we define some tools functions.

8.1.1 __footmisx_noHref:n

First we define a pointer to \ref* if hyperref is loaded or \ref if not loaded.

```

\__footmisx_noHref:n \@_noHref:n {\label}
759 \ExplSyntaxOn
760 \cs_new_nopar:Nn \__footmisx_noHref:n {\ref{#1}}

```

We run this code now (update) and eventually at end of hyperref file

```

\__footmisx_define__footmisx_noHref: \__footmisx_define__footmisx_noHref:
Define \__footmisx_noHref:n to something sensible.
761 \tl_const:Nn \__footmisx_define__footmisx_noHref: {
762   \@ifpackageloaded{hyperref}{
763     \cs_gset:Nn \__footmisx_noHref:n {\ref*{#1}}
764   }
765   {
766     \cs_gset:Nn \__footmisx_noHref:n {\ref{#1}}
767   }
768 }

```

Now redefine this function at end of hyperref thus allowing independant order of loading between hyperref and footmisx

```

769 \__footmisx_AtEndOfPackageFile_and_now:NN {hyperref}{
770   \__footmisx_define__footmisx_noHref: {}
771 }

```

8.2 __footmisx_makefnmarklink

This function create a footnote mark including an a link if hyperref is loaded.

```

\__footmisx_@makefnmarklink:n \__footmisx_@makefnmarklink:n {\marktoreset}

```

Typeset a footnote mark including link. Will be overridden by a hook later.

```

772 \cs_new_nopar:Nn \__footmisx_@makefnmarklink:n {\textfootmark{#1}}
We define the hyperref hook run when hyperref is not loaded

```

```

\__footmisx_noref_@makefnmarklink \__footmisx_noref_@makefnmarklink {\marktoreset}

```

Set a mark anchor in case of hyperref

```

773 \cs_new_eq:NN \__footmisx_noref_@makefnmarklink:n \__footmisx_@makefnmarklink:n
    We define the hyperref hook

```

```

\__footmisx_hyperref_@makefnmarklink:n \__footmisx_hyperref_@makefnmarklink:n {\marktoreset}

```

Set a mark anchor in case of hyperref

```

774 \cs_new_nopar:Nn \__footmisx_hyperref_@makefnmarklink:n {
775   \__footmisx_noref_@makefnmarklink:n {
776     \stepcounter{Hfootnote}%
777     \global\let\Hy@saved@currentHref\@currentHref
778     \hyper@makecurrent{Hfootnote}%
779     \global\let\Hy@footnote@currentHref\@currentHref
780     \global\let\@currentHref\Hy@saved@currentHref
781     \hyper@linkstart{link}{\Hy@footnote@currentHref}%
782     #1
783     \hyper@linkend
784   }
785 }

```

We run this hook now, and at end of hyperref package

```

\__footmisx_define__footmisx_@makefnmarklink: \__footmisx_define__footmisx_@makefnmarklink:

```

Define __footmisx_@makefnmarklink to something sensible.

```

786 \tl_const:Nn \__footmisx_define__footmisx_@makefnmarklink: {
787   \@ifpackageloaded{hyperref}{
788     \ifHy@hyperfootnotes
789       \cs_gset_eq:NN \__footmisx_@makefnmarklink:n \__footmisx_hyperref_@makefnmarklink:n
790     \else
791       \cs_gset_eq:NN \__footmisx_@makefnmarklink:n \__footmisx_noref_@makefnmarklink:n
792     \fi
793   }
794   {
795     \cs_gset_eq:NN \__footmisx_@makefnmarklink:n \__footmisx_noref_@makefnmarklink:n
796   }
797 }

```

Now run the hook

```

798 \__footmisx_AtEndOfPackageFile_and_now:NN {hyperref} {
799   \__footmisx_define__footmisx_@makefnmarklink: {}
800 }

```

Define now a xparse helper:

```

\__footmisx_@makefnmarklink_stared:Nn \__footmisx_@makefnmarklink_stared:Nn {\value or -novalue-}
\__footmisx_@makefnmarklink_stared:Nn {\mark}

```

If first argument is defined call no hyperlink version, else call hyperlink version.

```

801 \cs_new_nopar:Nn \__footmisx_@makefnmarklink_stared:Nn {
802   \IfBooleanTF{#1}
803     {\__footmisx_noref_@makefnmarklink:n{#2}}
804     {\__footmisx_@makefnmarklink:n{#2}}
805 }

```

Now define a the xparse handler

```

\__footmisx_makefnmarklink \@_makefnmarklink[*][\langle mark=\@thefnmark\rangle]
806 \DeclareDocumentCommand{\__footmisx_makefnmarklink}
807   {sO{\@thefnmark}}
808   {\__footmisx_@makefnmarklink_stared:Nn{#1}{#2}}
809 }

```

8.3 \@makefnmark

\makefnmark is unfixable, it is used both for a link and a target. We try our best and follow hyperref. By default \@makefnmark is a link (using the same approach than hyperref).

We begin by saving the original \@makefnmark.

```

\__footmisx_orig_@makefnmark \__footmisx_orig_@makefnmark
810 \cs_set_eq:NN \__footmisx_orig_@makefnmark \@makefnmark
Now we redefine \@makefnmark in compatible fashion

```

```

\@makefnmark \@_makefnmarklink[*][\langle mark=\@thefnmark\rangle]
811 \cs_set_eq:NN \@makefnmark \__footmisx_makefnmarklink

```

8.4 \@footnotemark

Change \@footnotemark to allow a star version (no hyperref) and an optional number.

```

\__footmisx_@footnotemark_stared:Nn \__footmisx_@footnotemark_stared:Nn {\langle value or -novalue-\rangle}
{\langle mark\rangle}

```

If first argument is defined call no hyperlink version, else call hyperlink version.

```

812 \cs_new_nopar:Nn \__footmisx_@footnotemark_stared:Nn {
813   \leavevmode
814   \mode_if_horizontal:TF
815   {
816     \edef\@x@sf{\the\spacefactor}
817     \FN@mfb@check
818     \nobreak
819   }
820   {}
821   \__footmisx_@makefnmarklink_stared:Nn{#1}{#2}
822   \FN@mfb@prepare
823   \mode_if_horizontal:TF
824   {\spacefactor\@x@sf}
825   {}
826   \relax
827 }

```

```

\__footmisx_@footnotemark \__footmisx_@footnotemark[*] [\<mark=\@thefnmark\>]
This a package specific compatible \@footnotemark.
828 \DeclareDocumentCommand{\__footmisx_@footnotemark}{sO{\@thefnmark}}{
829   \__footmisx_@footnotemark_stared:Nn{#1}{#2}
830 }

```

```

\__footmisx_define__footmisx_@makefnmarklink: \__footmisx_define__footmisx_@makefnmarklink:
Define \@footnotemark to \__footmisx_@footnotemark.
831 \tl_const:Nn \__footmisx_define_@footnotemark_tl: {
832   \cs_gset_eq:NN \@footnotemark \__footmisx_@footnotemark
833 }

```

```

\@footnotemark \@footnotemark[*] [\<mark=\@thefnmark\>]
This \@footnotemark extended to accept star and an optionnal arg.
834 \__footmisx_define_@footnotemark_tl:
835 \___footmisx_disablepatchinginpackage:cN{\@footnotemark}{hyperref}

```

8.5 \footnotemark

8.6 \@footnotetext

8.6.1 Hooks

```

\g__footmisx_footnotetextbeginhook_tl

```

This is the hook run at begining of a footnote. Used for instance by `setspace` pacakge to reset spacing.

```
836 \tl_new:N \g__footmisx_footnotetextbeginhook_tl
```

8.6.2 *setspace* package

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:

```

\g__footmisx_footnotetextbeginhook_setspace_tl

```

Hook run by `setspace` code at the beginning of `footnotetext`.

```
837 \tl_new:N \g__footmisx_footnotetextbeginhook_setspace_tl
```

Now we put `\g__footmisx_footnotetextbeginhook_setspace_tl` inside `\g__footmisx_footnotetextbeginhook_tl`.

```
838 \tl_put_right:Nn {\g__footmisx_footnotetextbeginhook_tl}
839   {\g__footmisx_footnotetextbeginhook_setspace_tl}
```

Now define the hook used to detect `setspace` like feature.

```

840 \tl_const:Nn \g__footmisx_definesetspacehook_tl: {
841   \@ifpackageloaded{setspace}{%
842     \@ifclassloaded{memoir}{%
843       \tl_gset:Nn {\g__footmisx_footnotetextbeginhook_setspace_tl}

```

```

844     {\cs_set_eq:NN \baselinestretch \m@m@singlespace}
845   }{%
846     \tl_gset:Nn {\g__footmisx_footnotetextbeginhook_setspace_tl}
847     {\cs_set_eq:NN \baselinestretch \setspace@singlespace}
848   }%
849   }{
850     \tl_gset:Nn {\g__footmisx_footnotetextbeginhook_setspace_tl} {}
851   }
852 }

```

We run this hook only if needed and to sensible default.

```

853 \__footmisx_AtEndOfPackageFile_and_now:NN {setspace} {
854   \g__footmisx_definesetspacehook_tl:
855 }

```

Now we disable setspace stuff.

```

856 \__footmisx_disablepatchinginpackage:cN{@footnotetext}{setspace}
857 \__footmisx_disablepatchinginpackage:cN{@mpfootnotetext}{setspace}

```

8.6.3 Margin footnote command

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

```

858 \ExplSyntaxOn
859 \cs_new:Nn \__footmisx_marginpar@footnotetext:N {
860   \marginpar{
861     % hook at beginning
862     \g__footmisx_footnotetextbeginhook_tl
863     \reset@font\footnotesize
864     \protected@edef\@currentlabel{\csname p@footnote\endcsname\@thefnmark}%
865     \color_group_begin:
866     \@makefntext{%
867       \ignorespaces#1%
868     }%
869     \color_group_end:
870   }%
871   \FN@m@f@prepare
872 }
873 %

```

8.6.4 Paragraph footnote command

```

874 \cs_new:Nn \__footmisx_par@footnotetext:N {
875   \insert\footins{%
876     % hook at beginning
877     \g__footmisx_footnotetextbeginhook_tl
878     \reset@font\footnotesize
879     \protected@edef\@currentlabel{\csname p@footnote\endcsname\@thefnmark}%
880     \interlinepenalty\interfootnotelinepenalty
881     \splittopskip\footnotesep
882     \splitmaxdepth \dp\strutbox
883     \floatingpenalty\@MM
884     \hsize\columnwidth
885     \@parboxrestore
886     \color_group_begin:

```

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

```
887 \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.

```
888 \makefntext{\ignorespaces#1\strut
```

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

```
889 \penalty-10\relax
890 \hskip\footglue
891 }% end of \makefntext parameter
892 }% end of \hbox
893 \dp\FN@tempboxa=Opt
894 \ht\FN@tempboxa=\dimexpr\wd\FN@tempboxa *
895 \footnotebaselineskip / \columnwidth\relax
896 \box\FN@tempboxa
897 \color_group_end:
898 }%
899 \FN@mfb@prepare
900 }
```

8.6.5 Normal footnote

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...

```
901 \cs_new:Nn \__footmisx_normal@footnotetext:N {
902   \insert\footins{%
903     \g__footmisx_footnotetextbeginhook_tl
904     \reset@font\footnotesize
905     \protected@edef\@currentlabel{\csname p@footnote\endcsname\@thefnmark}%
906     \interlinepenalty\interfootnotelinepenalty
907     \splittopskip\footnotesep
908     \splitmaxdepth \dp\strutbox
909     \floatingpenalty\@MM
910     \hsize\columnwidth
911     \@parboxrestore
912     \color_group_begin:
913     \makefntext{%
914       \rule\z@\footnotesep
915       \ignorespaces#1\@finalstrut\strutbox
916     }%
917     \color_group_end:
918   }%
919   \FN@mfb@prepare
920 }%
921 %
922 % \begin{macrocode}
923 \iffN@para
924 \cs_set_eq:NN \@footmisxnotetext \__footmisx_par@footnotetext:N
925 \else
926 \iffN@sidefn
927 \cs_set_eq:NN \@footmisxnotetext \__footmisx_marginpar@footnotetext:N
928 \else
```

```

929 \cs_gset_eq:NN \@footmisxnotetext \__footmisx_normal@footnotetext:N
930 \fi
931 \fi
932 \ExplSyntaxOff

```

Now define `\@footnotetext` to be `\@footmisxnotetext`. If `setspace` was loaded before `hyperref` we revert the change by using our code as `hypperef` hook.

```

933 \ExplSyntaxOn
934 \___footmisx_AtEndOfPackageFile_and_now:NN {hyperref} {
935   \@ifpackageloaded{hyperref}
936   {
937     \cs_gset_eq:cc {H @ @ footnotetext} {@footmisxnotetext}
938   }{
939     \cs_gset_eq:cc {@footnotetext} {@footmisxnotetext}
940   }
941 }
942 \ExplSyntaxOff

```

9 Other miscellaneous commands

9.1 Footnote mark style

`\textfootmark` Syntax `\textfootmark{<text>}`

Typographically set the text in the same style than a raw footnote mark (it is in fact a generic version of `\@makefnmark`)

```

943 \ExplSyntaxOn
944 \NewDocumentCommand{\textfootmark}{m}
945   {\hbox{\@textsuperscript{\normalfont#1}}}
946 \ExplSyntaxOff

```

(End definition for `\textfootmark`. This function is documented on page ??.)

9.2 Footnote references

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”.)

```

947 \ExplSyntaxOn
948 \cs_new_nopar:Nn \__footmisx_footref_stared:Nn {
949   \textfootmark{
950     \IfBooleanTF{#1}
951       {\__footmisx_noHref:n{#2}}
952       {\ref{#2}}
953   }
954 }
955 \DeclareDocumentCommand {\__footmisx_footref} {s m}
956 {
957   \__footmisx_footref_stared:Nn{#1}{#2}

```

```

958 }
959 \ExplSyntaxOff
We run hook now and at end of hyperref
960 \ExplSyntaxOn
961 \___footmisx_AtEndOfPackageFile_and_now:NN {hyperref} {\cs_gset_eq:NN \footref \___footmisx_f
962 \ExplSyntaxOff

```

9.3 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.

```

963 \ExplSyntaxOn
964 \DeclareDocumentCommand\mpfootnotemark{o}
965 {
966   \IfValueTF {#1}
967   {
968     \group_begin:
969     \cs_set:cpn {c@\@mpfn} {#1}
970     \unrestored@protected@xdef\@thefnmark{\thempfn}
971     \group_end:
972   }
973   {
974     \stepcounter\@mpfn
975     \protected@xdef\@thefnmark{\thempfn}
976   }
977   \@footnotemark
978 }
979 \ExplSyntaxOff

```

(End definition for \mpfootnotemark. This function is documented on page ??.)

```

980 \endinput
981 \</package>

```