

# The latex-lab-footnotes code\*

Frank Mittelbach, L<sup>A</sup>T<sub>E</sub>X Project

June 3, 2024

## Abstract

*to be written*

## Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
1.1	Configuration methods . . . . .	3
<b>2</b>	<b>Sockets and hooks</b>	<b>3</b>
2.1	Formatting the mark in the main text . . . . .	3
2.1.1	Sockets . . . . .	3
2.1.2	Hooks for formatting the footnote mark in text . . . . .	3
2.1.3	Additional configuration possibilities . . . . .	4
2.2	Formatting the footnote text . . . . .	4
2.2.1	Sockets . . . . .	4
2.2.2	Hooks for formatting the footnote text . . . . .	6
2.2.3	Additional configuration possibilities . . . . .	7
2.3	Debugging sockets and hooks . . . . .	7
<b>3</b>	<b>Tagging and hyperlinking support</b>	<b>7</b>
3.1	Technical details for the tagging . . . . .	7
3.2	Requirements for links . . . . .	8
3.3	The algorithmus to connect marks and notes . . . . .	8
3.3.1	\footref . . . . .	9
3.3.2	\footnotemark after \footnotetext . . . . .	9
3.4	Links . . . . .	10
3.5	Implementation details regarding tagging . . . . .	10
3.6	Handling the mark . . . . .	10
3.7	Handling the footnotetext . . . . .	10
3.8	Footnotes in minipages . . . . .	10
<b>4</b>	<b>TODOs</b>	<b>11</b>

---

\*

<b>5</b>	<b>The Implementation</b>	<b>11</b>
5.1	File declaration	11
5.2	code not fully handled yet	11
5.3	Temporary variables	12
5.4	Public variables	12
5.5	Internal variables	13
5.6	Variants	14
5.7	Updating <code>\@thefnmark</code>	14
5.8	Hooks	14
5.9	Debugging code	15
5.10	The new <code>\@footnotemark</code> command	16
5.11	The new <code>\@footnotetext</code> command	17
5.12	The new <code>\@makefnmark</code> command	20
5.12.1	Making documents use the new <code>\@makefnmark</code>	21
5.13	Document-level commands	23
5.14	Firstaid for packages and classes	24
5.15	Kernel patches	24
5.15.1	memoir	24
5.15.2	setspace	25
5.15.3	hyperref	25
5.16	Tagging and hyperlink code	25
5.16.1	Rolemap for structure tags	25
5.16.2	Extending the label system	25
5.16.3	Storing and retrieving reference data	26
5.16.4	Enabling tagging and links for the mark command	27
5.16.5	The footnote text	28
<b>6</b>	<b>Reimplementing the <code>footmisc</code> package</b>	<b>30</b>
	<b>Index</b>	<b>42</b>

## 1 Introduction

This code reimplements the footnote interfaces for  $\text{\LaTeX}$  offering configurable methods for layout and functionality adjustments that avoid overwriting each other when used in classes as well as in packages (as far as possible — obviously some adjustments are mutually exclusive). This is achieved by providing a larger number of hooks (for areas where different packages/classes can easily coexist with their adjustments) and a number of sockets to which only one class or package can write to successfully (in case of multiple changes the last one wins). The latter are for special functionality, e.g., if footnote text is typeset as a single paragraph, it can't be configured the same time to be typeset vertically with one footnote below each other.

The interfaces are set up to support tagged PDF, but in order for this to work, all packages altering the footnote setup should use the interfaces provided here and not do it through the legacy methods (though there is some support for the latter as well, but it will not work in all cases).

## 1.1 Configuration methods

Historically, the footnote setup in L<sup>A</sup>T<sub>E</sub>X was done by providing definitions for `\@makefnmark` (format the footnote mark in running text and in front of the footnote text) and `\@makefntext` (formatting the footnote text and placing a mark in front of it).

There was a default definition for `\@makefnmark` in the format that was used by most document classes, but `\@makefntext` had to be defined in the class itself because the format didn't provide a default. As a result you will find definitions for the latter in all document classes and definitions for `\@makefnmark` only in very few.

Furthermore, to enable special footnote layouts or provide additional functionality a few packages (and a few classes) overwrote other internal commands of L<sup>A</sup>T<sub>E</sub>X's footnote mechanism. The commands affected in this way are mainly `\@footnotemark` and `\@footnotetext`. These overwrites could not be used in combination, so either the packages/classes had to be aware of being loaded together (which they sometimes did or tried to) or they would fail by overwriting each other unconditionally.

The present rewrite is an attempt to improve this situation, but of course, it will only work if all packages/classes make use of the new interfaces. Fortunately, the number of problematical packages altering these internal commands are fairly small so arranging for updates is a realistic goal — to achieve properly tagged PDF it is a requirement.

## 2 Sockets and hooks

We use sockets for those parts that can be controlled only by one package or by the kernel and hooks for places where it may be possible that several packages or the document class adds code (typically declarations such as font changes, etc.).

Note that sockets are of interest only to very few specialized packages, mainly `footmisc`, and packages providing similar functionality—the current documentation is therefore fairly sketchy.

In contrast the hooks are of interest to many classes to provide their layout alterations in a way that it works smoothly with other packages handling aspects of footnote formatting.

### 2.1 Formatting the mark in the main text

This implements formatting the mark<sup>1</sup> and its relation to surrounding text, e.g., if several marks appear in the same place, etc.

#### 2.1.1 Sockets

None: everything is implemented through a single definition for `\@footnotemark` that offers a number of hooks that can be used by packages to implement handling of multiple marks and the formatting of marks.

#### 2.1.2 Hooks for formatting the footnote mark in text

The hooks to customize the marks in the text are the following:

---

<sup>1</sup>Like this one.

### **fnmark/before**

Executed at the very beginning of `\footnotemark`. Currently there are two packages (`bibarts` and `chextras`) that prepend material at this point (not necessarily correctly, e.g., they do not all check that they are in horizontal mode).

This hook is paired with hook `fnmark/after`.

### **fnmark**

Executed in horizontal mode and after the current space factor has been saved away for reuse. This is where currently code for multiple marks does its preparation (as done by `footmisc` and others).

The hook is only executed in `hmode`, i.e., not if the mark is generated in math — maybe that means the multiple handling should happen later?

After the hook a `\nobreak` is executed, so any “material” added in the hook is tied to the following mark unless it contains its own permissible penalty.

### **fnmark/begin**

This hook is executed directly in front of the typeset mark. This is the place where `hyperref` would have added part of its code, i.e., after the `\nobreak` mentioned above. With the integration of hyperlinks in the tagging code this hook may not be necessary at all.

### **fnmark/end**

This hook is executed directly after the typeset mark. It is used by `memhfixc`, `sctrltr2`, and `footmisc`. Used, for example, to implement support for multiple marks in succession.

It is *not* a reversed hook.

### **fnmark/after**

This hook is executed at the very end of the `\footnotemark` command.

It is a reversed hook to pair with `fnmark/before`

## **2.1.3 Additional configuration possibilities**

The actual formatting is done through `\@makefnmark` — no special customization support for now.

## **2.2 Formatting the footnote text**

This implements the formatting of the footnote text the way it appears at the bottom of the page (default case), or possibly elsewhere, e.g. in the margin.

### **2.2.1 Sockets**

To cater for different layout configurations there are four sockets that can be set by a package or class but there should be only one per document setting them, i.e., if two packages/classes set them they are mutually incompatible (or rather the last one wins most likely). These are:

**fntext/process (1 argument)**

This socket receives all material that is to be processed (or stored) including color protection code and what have you. The `default` executes `\insert\footins`.

Available plugs are `default`, `side` (side notes), and `mp` (minipage).

**fntext/make (1 argument)**

This socket receives the  $\langle text \rangle$  as given in `\footnote` or `\footnotetext` in the document and adds formatting instructions to it.

The `default` plug runs `\@makefntext` which contains various hooks for customization. For most scenarios this is sufficient. However, when running all footnotes as a single paragraph at the bottom, then each footnote needs to be prepared prior to storing it with `\insert` and this socket allows running extra code to do that.

Available plugs are `default` and `para`.

**fntext/begin (no argument)**

The socket is executed near the start of the argument for the `fntext/make` socket. By `default` it adds a strut to the footnote material so that consecutive footnotes are properly spaced vertically. In some use cases this is not appropriate (e.g., when running all footnotes as a single paragraph) and so with this socket one can cancel the action or do something else instead.

Available plugs are `default` and `noop`.

**fntext/end (no argument)**

This socket is executed at the very end of the argument passed to socket `fntext/make`. By `default` it adds a final strut as long as we are still in horizontal mode (i.e., processing the footnote text paragraph). When running several footnotes in one paragraph some additional material (some horizontal glue) needs adding at this point which is done with the plug `para`.

Available plugs are `default`, `para`, and `noop`.

All standard plugs for the socket `fntext/make` run `\@makefntext` and this command contains two further sockets (unless it is overwritten by a legacy class):

**fntext/mark (0 arguments)**

This socket has no input arguments but uses `\@makefnmark` to typeset the mark in front of the footnote text. Its `default` uses code that examines the value of `\footnotemargin` and based on its setting typeset the mark in different ways:

- positive: typeset the mark in a box of that size
- zero: use `\llap` around the mark
- negative: use `\llap` but with a box of the given size negated inside
- `-\maxdimen`: just use `\@makefnmark`

For most cases this would be flexible enough, but if not then a class can define its own plug to specify the placement of the mark.

Available plugs are `default` and `noop` (no mark is produced).

### **fn`text`/text (1 argument)**

This socket manages the formatting of the footnote text (presented as an argument) once the mark has been typeset. In all cases we can think of this formatting is better configured via the available hooks described below, so the `default` just grabs the argument and processes it without any other action. It is really only there to allow for some fancy stuff that some design comes up with.

Available plugs are `identity` (default) and `noop`.

The above configuration points are sufficient to implement all commonly used footnote layouts assuming L-R typesetting. For R-L typesetting they or may or may not need some extension (though that is not clear right now).

## **2.2.2 Hooks for formatting the footnote text**

### **fn`text`/before**

Executed at the very beginning of `\footnotetext`. Currently there is one package (`linguex`) that prepends material at this point.

This hook is paired with hook `fnmark/after`.

### **fn`text`**

Executed at the beginning of the material passed to the first configuration point. Typically used to set any baseline stretch for the footnote text, e.g., by `setspace`, `footmisc`, `uathesis.cls` and others. Could be done in a later hook but is a bit more efficient here.

After the hook has run, the font is established, i.e., it can't be used to set a different font size.

### **fn`text`/para**

After the font is set (after the previous hook), some default paragraph parameters are set up including `\interlinepenalty`, `\hsize`, `\parindent` and a number of others, as some of them depend on the font size. Then the `fntext/para` is run which can overwrite the default. If one wants to change the font size, it is probably necessary to reset these other parameters too, e.g., `\parindent`, which can be done here.

Note: the socket `fntext/make` normally runs the command `\@makefntext` or some code that eventually runs this command, and this then produces the footnote mark in front of the formatted footnote text. In front of both the mark and the footnote text some classes have placed paragraph parameter adjustments in their redefinition of `\@makefntext`. However, there is no need to place it there it could equally well go into the `fntext/para` hook. We therefore do not provide another hook at this other point.

### **fn`text`/begin & fn`text`/end**

The footnote text itself is surrounded by the hooks `fntext/begin` and `fntext/end`. The two hooks are not paired as they are typically used independently.

### **fn`text`/after**

At the very end of `\footnotetext` we execute the hook `fntext/after` which is a reversed hook paired with `fntext/before`. Some packages, e.g., `linguex`, have code in that position.

### 2.2.3 Additional configuration possibilities

The formatting of the footnote mark in front of the footnote text is influenced by the setting of the `dimen` parameter `\footnotemargin`. By default its value is 1.8em in the current text font (or `-\maxdimen` when the `para` option is chosen). The following rules apply:

- If it has the value `-\maxdimen` then the mark is generated by `\@makefnmark`.
- Otherwise, if the value is negative then the mark is placed into an `\llap` left aligned in a box of size `-\footnotemargin`.
- If the value is zero an `\llap` is used without an inner box.
- If the value is greater zero (but less than `\maxdimen`) the mark is placed right aligned into a box of size `\footnotemargin`.
- The value `\maxdimen` is used as a marker to indicate that no value was given and that the default should be used, i.e. 1.8em or `-\maxdimen` depending on the chosen option.

## 2.3 Debugging sockets and hooks

For some rudimentary debugging we currently have `\DebugFNotesOn` (and `\DebugFNotesOff`). At the moment `\DebugFNotesOn` only shows the current settings for hooks and sockets related to the footnote code and then automatically turns itself off again.

## 3 Tagging and hyperlinking support

*TODO: this section needs work (and probably csnames changes)*

Footnotes consist of a *footnotemark* (short: mark) that is typically placed in the text as a superscript number like this<sup>1</sup>, and a *footnotetext* (short: note) that is placed at the bottom of the page. The *footnotetext* normally repeats at the begin the mark as a visual clue.

Tagging (and hyperlinking) has to connect the mark with the note. For the tagging code, we assume that every mark has exactly one associated note, and that every note is associated to at least one mark and can have more associated marks.

The mark doesn't need to be visible, e.g. the typesetted mark<sup>1-3</sup> denotes three marks, where the second is invisible. Tagging should produce here probably three `Lb1` structures (one without content), and an artifact for the range marker. If such a range is used, links can only point to the notes 1 and 3 and one has to suppress the linking for the second mark. This means that links and tagging are also related to the actual formatting of the footnote mark. In the following this problem is mostly ignored for now, but should not be forgotten and handled later.

### 3.1 Technical details for the tagging

The following sockets are set up for kernel use, when doing tagging:

**tagsupport/fnmark (1 argument)**

*TODO: describe and decide on names*

`tagsupport/fntext/begin` (no argument)

`tagsupport/fntext/end` (no argument)

`tagsupport/fntext/mark` (1 argument)

`tagsupport/fntext/text` (1 argument)

The *footnotemark* should create a `/Lb1` structure<sup>2</sup> that should contain a `/Ref` entry pointing to the structure of the *footnotetext*.

The *footnotetext* should create a `/FENote`<sup>3</sup> structure with a `/Ref` entry pointing to the structures of *all* marks related to the note. The mark at the begin of the note is in a `/Lb1`<sup>4</sup> structure but has to fulfil no special requirements.

Structure objects and the underlying properties used by the tagging code are initialized when the structure is opened. This means that one can not directly add data to a future structure but as structure objects are written at the end of the document it is possible to update `/Ref` entries in an end document hook.

So tagging has to solve two problems:

- the mark and the footnote text must be surrounded by the correct structure and marked content commands. This is not trivial as there are various layouts (bottom, marginpar, minipage) and the tagging from the automatic paratagging must be taken into account if one want to avoid faulty nesting.
- It must detect which marks are related to which notes so that it can setup the `/Ref` cross-references.

## 3.2 Requirements for links

Links should go from the mark to the note. Sometimes it has been requested that links go back too, but as there can be more than one mark connected to a note it is not clear how to decide to which mark it should go. Using the keys from the PDF viewer to go back is normally better.

Links are closely related to the references stored in the `/Ref` entry of a mark and so are handled in the code together with them. But there are subtle technical differences to take care of as links and destinations are whatsits and so must be created at the correct time.

It should be possible to suppress the links both globally and locally.<sup>5</sup>

## 3.3 The algorithmus to connect marks and notes

The connection is made by comparing the value of `\@thefnmark`.

The standard mark commands (`\footnotemark` and `\footnote`) store the current value of `\@thefnmark` with their own structure number as a key in a property.

A following `\footnotetext` compares its own `\@thefnmark` with the values in the prop. If there is a match it stores the structure numbers and removes the entries from the properties (so in a normal document the property will never contain more than a few entries).

---

<sup>2</sup>to make it easier to identify the role we use `/footnotemark` which we rolemap to `/Lb1`

<sup>3</sup>We tag it as `/footnote` and role map it.

<sup>4</sup>We tag it as `/footnotelabel`.

<sup>5</sup>Currently `hyperref` only offers the option to suppress the footnote links globally with the option `hyperfootnotes=false`. To suppress them locally only the `NoHyper` environment is provided.



This works well as long as the `\footnotemark` commands are issued before the `\footnotetext` and as long as nothing unusual is done to `\@thefnmark`. It also works if a document uses more than one footnote series as long as they have distinct numbering systems, but in case a distinction is needed it is possible to define a new class with its own data structure and to switch locally to use this class. The following three commands are used for this.

The default class uses the name `default`

---

```
\fnote_class_new:nn \fnote_class_new:nn{<name>}{<key/value option>}
```

This declaration sets up the needed data structure. Currently this only consists of a property list which is used to store and manage the mark values. There are no options yet.

---

```
\fnote_mark_gput:nn \fnote_mark_gput:nn{<mark>}{<class name>}
```

---

```
\fnote_mark_gput:(no|oo)
```

This command stores the current structure number as key and the `<mark>` as value in the property list associated with the `<class name>`.

---

```
\fnote_mark_gpop_all:nnN \fnote_mark_gpop_all:nnN{<mark>}{<class name>}{<sequence>}
```

This command stores all the keys/structure numbers whose value in the property list for `<class name>` are equal to `<mark>` into the sequence `<sequence>` and then removes them from the property list. The content of the sequence can then be used to create link targets and references.

### 3.3.1 `\footref`

`\footref` use internally the same command to set the mark as `\footnotemark`, it only defines `\@thefnmark` differently. This `\@thefnmark` is not suitable for the method described above, as it contains a reference command it can't be used to match a note, also `\footref` can be used after the note has already been set. `\footref` disables therefore the automatic detection.

Instead the `\label` command is extended in the `\footnotetext` command to also store the structure number and `\footref` retrieves this number to setup the reference and the link.

The structure related to the `\footref` is added to the end of the `/Ref` array of the note and so the `/Ref` array doesn't necessarily reflect the order of the marks in the document. It would probably be possible to change this, but it is not clear if it actually matters and so it worth the additional coding and processing.

### 3.3.2 `\footnotemark` after `\footnotetext`

The automatic detection doesn't work if a `\footnotemark` is issued after the `\footnotetext` it refers to. There will be no error, but neither the link nor the `/Ref` will connect both.

The simple way to handle this is to use a label and `\footref`:

```
\footnotetext{\label{fn:a}text} ... \footref{fn:a}
```

An alternative would be to extend the syntax of `\footnotemark` and `\footnotetext` to allow to add a label which can then be used. For example

```
\footnotetext[label=fn:a]{text} ... \footnotemark[label=fn:a]
```

As both have already an optional argument, that requires the optional argument extension.

### 3.4 Links

The structure numbers detected for the `/Ref` are also used for links: even if tagging is not activated the tagging commands are defined through the `tagpdf-base` package and the structure commands increase the structure counter and this info can be used.

A `\footnotetext` creates a bunch of destinations (in most cases this sums up to two destinations): one for every structure number in the `/Ref` (used as target by the mark commands) and one for the structure number of the `footnotetext` itself (used as target by `\footrefs` commands).

### 3.5 Implementation details regarding tagging

#### 3.6 Handling the mark

The mark in the text is handled by assigning an appropriate plug to the socket `taggsupport/fnmark`. It takes one argument, `\@makefnmark`, the command which formats the mark, and surrounds it by link and tagging commands. At the point where the socket is executed, `\@thefnmark` has already been defined and can be used to setup the reference detections.

#### 3.7 Handling the footnotetext

The main part is done by assigning a different plug to socket `taggsupport/fntext/begin` and `taggsupport/fntext/end` surrounding the footnote text. These sockets are used to start and end the structure and attempt to detect to which mark the note is related.

The actual typesetting of the note text is done by `\fnote_makefntext:n` (or its L<sup>A</sup>T<sub>E</sub>X<sub>2 $\epsilon$</sub>  name `\@makefntext`). In the new implementation this contains two further kernel sockets for tagging: `taggsupport/fntext/mark` and `taggsupport/fntext/text`. They get plugs assigned that add the tagging commands around note mark and note text.

#### 3.8 Footnotes in minipages

In minipages the `\footnote` command uses a special marker (small italic letters by default) and puts the footnote text at the bottom of the box. The `\footnotemark` command uses the standard footnote counter and marker (and so typically creates a superscript number). It is meant to be used with a `\footnotetext` *outside* the minipage to create a footnote mark which refers to a footnote text at the bottom of the page. This means to repeat a footnote marker in a minipage you should use the `\footref` command.

Tagging works quite similar to normal footnotes if the new definition is used and if the minipage code is changed to use the new configuration point. The main problem here is currently the tagging of the minipage itself.

## 4 TODOs

- Special formatting of footnote marks in the text, e.g. if ranges or commas are used require special care as they should normally mark up such text as artifacts and perhaps have to insert empty structures to represent an invisible mark. This must be coordinated with the relevant packages and classes.
- manyfoot doesn't work correctly and must be analyzed.
- memoir is not supported at all and errors when the code tries to patch `\@makefn`text.

*To be documented*

## 5 The Implementation

All this is very rough and misses a lot of documentation.

```
1 <*kernel>
2 <@@=fnote>
```

### 5.1 File declaration

```
3 \ProvidesFile{latex-lab-footnotes.ltx}
4           [\ltxlabfootnotedate\space v\ltxlabfootnoteversion\space
5           changes to the footnote interfaces]
```

### 5.2 code not fully handled yet

```
6 %
7 % latex.ltx
8 % not looked at yet
9 % \@mpfootnotetext is probably no longer needed, or only to support other
10 % classes and package. See below about the minipage code.
11 %
12 % \long\def\@mpfootnotetext#1{%
13 %   \global\setbox\@mpfootins\vbox{%
14 %     \unvbox\@mpfootins
15 %     \reset@font\footnotesize
16 %     \hsize\columnwidth
17 %     \@parboxrestore
18 %     \def\@currentcounter{mpfootnote}%
19 %     \protected@edef\@currentlabel
20 %       {\csname p@mpfootnote\endcsname\@thefnmark}%
21 %     \color@begingroup
22 %       \@makefn
```

```
text{%
23 %         \rule\z@\footnotesep\ignorespaces#1\@finalstrut\strutbox}%
24 %     \par
25 %     \color@endgroup}}
26 % =====
27 % used by the minipage footnote code.
28 %
29 % \def\@mpfn{footnote}
30 % \def\thempfn{\thefootnote}
31 % =====
```

```

32 % this perhaps need some configuration options.
33 %
34 %\def\@makefnmark{\hbox{\@textsuperscript{\normalfont\@thefnmark}}}}
35 %
36 % =====
37 %% alterations not covered:
38 %
39 % ./arabtex/afoot.sty --- too different (and probably too old)
40 %
41 % =====
42 % alterations of footnotetext not covered:
43 %
44 % ./revtex4-1/revtex4-1.cls ./revtex/ltxutil.sty ./revtex/revtex4-2.cls ... (need analysis)
45 % ./bigfoot/bigfoot.sty
46 %
47 % memoir needs checking too
48 %
49 % =====
50 %
51 % use of kerns to mark h-mode positions (unit sp)
52 %
53 % 1 = CJK
54 % 2 = CJK
55 % 3 = multiple footnotes (footmisc, koma, eledmac, tufte, memoir,
56 %   parnotes, sidenotes)
57 % 3 = outer kern in letter spacing (letterspace)
58 % 3 = beginning of list (examdesign.cls)
59 % 4 = CJK pigin
60 % 5 = CJK ruby
61
62 % 1-4 = polyglossia for korean
63 %

```

---

```
64 \ExplSyntaxOn
```

### 5.3 Temporary variables

```
65 \prop_new:N \l__fnote_tmpa_prop
66 \tl_new:N \l__fnote_tmpa_tl
```

### 5.4 Public variables

A footnote mark will store its structure number (key) and the expanded `\@thefnmark` in this prop so that a following note can retrieve this info if needed. It is possible to use more than one footnote series (type) if needed (if different footnotes/note use the same numbering system). If this command is changed an accompanying property must be created

```
\l_fnote_type_tl
```

```
67 \tl_new:N \l_fnote_type_tl
68 \tl_set:Nn \l_fnote_type_tl {default}
```

*(End of definition for `\l_fnote_type_tl`. This function is documented on page ??.)*

It must be possible to suppress the hyperlinking, both locally and globally. `hyperref`'s `hyperfootnotes` option should set the boolean.

`\l_fnote_link_bool`

```
69 \bool_new:N \l_fnote_link_bool
70 \bool_set_true:N \l_fnote_link_bool
```

*(End of definition for \l\_fnote\_link\_bool. This function is documented on page ??.)*

A hyperlink should have an changeable link type. This can be e.g. used to change the color or the border.

`\l_fnote_link_type_tl`

```
71 \tl_new:N \l_fnote_link_type_tl
72 \tl_set:Nn \l_fnote_link_type_tl {link}
```

*(End of definition for \l\_fnote\_link\_type\_tl. This function is documented on page ??.)*

## 5.5 Internal variables

`\l__fnote_linktarget_tl` This command stores the name of a linktarget/destination when needed

```
73 \tl_new:N \l__fnote_linktarget_tl
```

*(End of definition for \l\_\_fnote\_linktarget\_tl.)*

`\l__fnote_currentlabel_tl` This command is used to pass a label name around.

```
74 \tl_new:N \l__fnote_currentlabel_tl
```

*(End of definition for \l\_\_fnote\_currentlabel\_tl.)*

`\l__fnote_currentrefs_seq` This sequence stores the list of reference of a note

```
75 \seq_new:N \l__fnote_currentrefs_seq
```

*(End of definition for \l\_\_fnote\_currentrefs\_seq.)*

The connection between the mark(s) in the text and the note is either deduced automatically or done through an label. The default is automatic, but we must be able to suppress it. For this we use a boolean.

`\l__fnote_autodetect_bool`

```
76 \bool_new:N \l__fnote_autodetect_bool
77 \bool_set_true:N \l__fnote_autodetect_bool
```

*(End of definition for \l\_\_fnote\_autodetect\_bool.)*

This is used to pass the structure number of the note around, e.g. to a label inside the note.

```
78 \tl_new:N \l__fnote_currentstruct_tl
79 \tl_set:Nn \l__fnote_currentstruct_tl {1}
```

## 5.6 Variants

```

80 \cs_generate_variant:Nn \hook_gput_code:nnn{nne}
81 \cs_generate_variant:Nn \tag_struct_use:n {e}

```

## 5.7 Updating \@thefnmark

`\fnote_step_fnmark:nn` This command updates \@thefnmark. The first argument is an optional integer expression, the second a counter name. If the optional argument is not given it steps the counter.

```

82 \cs_new_protected:Npn \fnote_step_fnmark:nn #1#2 {
83   \tl_if_novalue:nTF {#1}
84     {
85       \stepcounter {#2}
86       \protected@xdef \@thefnmark { \use:c { the#2 } }
87     }
88     {
89       \group_begin:

```

Note that this is a local assignment even though L<sup>A</sup>T<sub>E</sub>X counters are normally globally changed. This is the way it was in 2e and so far we haven't changed it. The alternative would be to store the current value and restore it after \@thefnmark is altered.

```

90         \int_set:cn { c@#2 }{ #1 }
91         \unrestored@protected@xdef \@thefnmark { \use:c { the#2 } }
92       \group_end:
93     }
94 }

```

*(End of definition for \fnote\_step\_fnmark:nn. This function is documented on page ??.)*

`\fnote_set_fnmark:nn` This is similar to the previous command, but it doesn't step the counter but use the current value.

```

95 \cs_new_protected:Npn \fnote_set_fnmark:nn #1#2 {
96   \tl_if_novalue:nTF {#1}
97     {
98       \protected@xdef \@thefnmark { \use:c { the#2 } }
99     }
100    {
101      \group_begin:
102        \int_set:cn { c@#2 }{ #1 }
103        \unrestored@protected@xdef \@thefnmark { \use:c { the#2 } }
104      \group_end:
105    }
106 }

```

*(End of definition for \fnote\_set\_fnmark:nn. This function is documented on page ??.)*

## 5.8 Hooks

```

fnmark/before (hook) Hooks in the footnotemark command.
fnmark/after (hook) 107 \NewMirroredHookPair{fnmark/before}{fnmark/after}
fnmark (hook)       108 \NewHook{fnmark}
fnmark/begin (hook) 109 \NewHook{fnmark/begin}
fnmark/end (hook)   110 \NewHook{fnmark/end}

```

```

fntext/before (hook) Hooks in the footnotetext command:
fntext/after (hook) 111 \NewMirroredHookPair{fntext/before}{fntext/after}
    fntext (hook) 112 \NewHook{fntext}
fntext/begin (hook) 113 \NewHook{fntext/para}
    fntext/end (hook) 114 \NewHook{fntext/begin}
fntext/para (hook) 115 \NewHook{fntext/end}

```

## 5.9 Debugging code

The debugging code is just temporary

```

116 \bool_new:N          \g_fnote_debug_bool

\DebugFNotesOn
\DebugFNotesOff 117 \cs_new_protected:Npn \DebugFNotesOn { \bool_gset_true:N \g_fnote_debug_bool }
118 \cs_new_protected:Npn \DebugFNotesOff { \bool_gset_false:N \g_fnote_debug_bool }

```

(End of definition for `\DebugFNotesOn` and `\DebugFNotesOff`. These functions are documented on page ??.)

We log the hooks in the footnote mark command, but only once

```

119 \cs_new_protected:Npn \__fnote_debug_footnotemark:
120 {
121   \bool_if:NT \g_fnote_debug_bool
122     {
123       \hook_log:n {fnmark/before}
124       \hook_log:n {fnmark}
125       \hook_log:n {fnmark/begin}
126       \hook_log:n {fnmark/end}
127       \hook_log:n {fnmark/after}
128       \cs_gset_eq:NN \__fnote_debug_footnotemark: \prg_do_nothing:
129     }
130 }

```

Similar for the footnotetext

```

131 \cs_new_protected:Npn \__fnote_debug_footnotetext:
132 {
133   \bool_if:NT \g_fnote_debug_bool
134     {
135       \socket_log:n {fntext/process}
136       \socket_log:n {fntext/make}
137       \socket_log:n {fntext/begin}
138       \socket_log:n {fntext/end}
139
140       \socket_log:n {fntext/mark}
141       \socket_log:n {fntext/text}
142
143       \socket_log:n {tagssupport/fnmark}
144       \socket_log:n {tagssupport/fntext/begin}
145       \socket_log:n {tagssupport/fntext/end}
146       \socket_log:n {tagssupport/fntext/mark}
147       \socket_log:n {tagssupport/fntext/text}
148
149       \hook_log:n {fntext/before}
150       \hook_log:n {fntext}
151       \hook_log:n {fntext/para}
152       \hook_log:n {fntext/begin}

```

```

150     \hook_log:n {fntext/end}
151     \hook_log:n {fntext/after}

```

Show the info only once (if at all).

```

152     \cs_gset_eq:NN \_fnote_debug_footnotetext: \prg_do_nothing:
153   }
154 }

```

## 5.10 The new \@footnotemark command

`\fnote_footnotemark:` This is the main command which will replace `\@footnotemark`.

```

155 \cs_new_protected:Npn \fnote_footnotemark: {
156   \_fnote_debug_footnotemark:
157   %-----
158   % bibarts
159   % chextras --- actually in the wrong place does an \unskip
160   \hook_use:n {fnmark/before}
161   %-----
162   \leavevmode
163   \ifhmode
164     \edef\@x@sf{\the\spacefactor}
165   %-----
166   % bxjsja-minimal.def --- what they do could be done at ‘‘bibarts’’
167   %                               (a bit less efficient)
168   % memhfixc.sty
169   % footmisc.sty
170   \hook_use:n {fnmark}
171   %-----
172   \nobreak
173   \fi
174   %-----
175   % hyperref.sty
176   \hook_use:n {fnmark/begin}
177   %-----

```

The kernel socket for tagging. It picks up `\@makefnmark` as its argument and if tagging is not active it contains the identity plug.

```

178   \socket_use:nn {taggsupport/fnmark} \@makefnmark
179   %-----

```

If a footnote mark is placed by its own then it should finish by executing the hook `fnmark/end`, resetting the space factor, and finishing with the hook `fnmark/after`. However, in a complete footnote these actions have to happen only after we have handled the footnote text (e.g., by placing it into an `\insert`). In such a situation `\_fnote_footmark_finish:` below does nothing and the action is carried out later.

```

180   \_fnote_footnotemark_finish:
181 }

```

*(End of definition for `\fnote_footnotemark:`. This function is documented on page ??.)*

`\_fnote_footnotemark_default_finish:` The default definition for `\_fnote_footnotemark_finish:` is called `\_fnote_footnotemark_default_finish:`

```

182 \cs_new_protected:Npn \_fnote_footnotemark_default_finish: {
183 % hyperref.sty

```



```

184 % memhfixc.sty --- could move fnmark/after
185 % scr1ttr2.cls --- could vanish if footmisc uses a hook
186 % footmisc.sty
187 \UseHook{fnmark/end}
188 %-----
189 \ifhmode
190 \spacefactor \@x@sf \relax
191 \fi
192 %
193 %-----
194 \UseHook{fnmark/after}
195 %-----
196 }
197 \cs_new_eq:NN \_fnote_footnotemark_finish: \_fnote_footnotemark_default_finish:

```

(End of definition for `\_fnote_footnotemark_default_finish:` and `\_fnote_footnotemark_finish:`.)

`tagsupport/fnmark (socket)` Not a public socket but reserved for tagging. By default it contains identity and is reassigned if tagging is active.

```
198 \NewSocket{tagsupport/fnmark}{1}
```

`\@footnotemark` Here we provide the traditional L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> name in case it is directly used in some legacy class.

```
199 \cs_set_eq:NN \@footnotemark \fnote_footnotemark:
```

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

## 5.11 The new `\@footnotetext` command

`\fnote_footnotetext:n`

```

200 \cs_new_protected:Npn \fnote_footnotetext:n #1 {
201 \_fnote_debug_footnotetext:
202 %-----
203 % ./linguex/linguex.sty
204 \hook_use:n {fntext/before}
205 %-----

```

Execute a kernel socket for tagging.

```

206 \socket_use:n {tagsupport/fntext/begin}
207 \socket_use:nn {fntext/process}
208 {
209 %-----
210 % resetting baselinestretch ... (could be done further down)
211 % ./uafthesis/uafthesis.cls
212 % ./setspace/setspace.sty
213 % ./footmisc/footmisc.sty (normal)
214 \hook_use:n {fntext}
215 %-----
216 \reset@font
217 \footnotesize
218 %-----
219 % some classes use a different font size, e.g.,
220 % ./nrc/nrc1.cls ./nrc/nrc2.cls
221 % but those could be done in fntext/para instead
222 %-----

```

In case of sidenotes the next settings are pointless, but as they do not hurt (except for the `\hsize` setting) and are needed for all other cases we make them here and overwrite them for side notes

```

223     \interlinepenalty\interfootnotelinepenalty
224     \splittopskip\footnotesep
225     \splitmaxdepth \dp\strutbox
226     \floatingpenalty \@MM
227     \hsize\columnwidth
228     \@parboxrestore
229     \parindent 1em      % typical default used in \@makefnstext moved up here
230     \def\@currentcounter{footnote}
231     \protected@edef \@currentlabel { \p@footnote \@thefnmark }
232 %-----
233 % for altering para parameters ...
234 % code for resphilosophica came earlier but it could go here.
235 % Has the advantage that one can also overwrite \cs{@currentcounter}
236 % and \cs{@currentlabel} is that is necessary.
237 %
238 % ./resphilosophica/resphilosophica.cls
239     \hook_use:n {fnstext/para}
240 %-----
241     \color@begingroup
242 %-----
243 % fnpara wants to replace \@makefnstext{...} and para and side
244 % option of footmisc etc too ...
245 % so we make this a socket, because only one action can be active:
246 %-----
247     \socket_use:nn {fnstext/make}
248     {
249 %-----
250 % ./resphilosophica/resphilosophica.cls
251 %-----
252     \socket_use:n {fnstext/begin}%
253 %-----
254 % bibarts
255 % fnbreak.sty
256     \hook_use:n {fnstext/begin}
257 %-----
258     \ignorespaces
259     #1
260 %-----
261 % bibarts
262 % fnbreak.sty
263     \hook_use:n {fnstext/end}
264 %-----
265     \socket_use:n {fnstext/end}
266     }
267     \par
268     \color@endgroup
269 }
270 %-----

```

The socket code (by default adding a strut) has to come *after* everything added into the hook above.

The corresponding kernel hook that ends the tagging structure if tagging is active.

```

271 \socket_use:n{tagsupport/fntext/end}
272 %-----
273 % ./linguex/linguex.sty
274 \hook_use:n {fntext/after}
275 %-----
276 }

```

(End of definition for `\fnote_footnotetext:n`. This function is documented on page ??.)

`fntext/process (socket)`

```

277 \NewSocket      {fntext/process}{1}
278 \NewSocketPlug{fntext/process}{default}{ \insert\footins {#1} }
279 \NewSocketPlug{fntext/process}{side}   { \marginpar {#1} }
280 \AssignSocketPlug{fntext/process}{default}

```

`fntext/make (socket)` This socket receives the `<text>` from the `\footnote` or `\footnotetext` and formats it.

```

281 \NewSocket      {fntext/make}{1}
282 \NewSocketPlug{fntext/make}{default}{ \@makefntext {#1} }

```

When running several footnotes together as a paragraph some additional work is necessary to unbox the individual footnotes recursively (see `TEXbook` algorithm in appendix D).

```

283 \NewSocketPlug{fntext/make}{para}
284 {
285   \setbox\FN@tempboxa\hbox{\@makefntext{#1}}%
286   \dp\FN@tempboxa\z@
287   \ht\FN@tempboxa
288   \dimexpr\wd\FN@tempboxa *%
289           \footnotebaselineskip /\columnwidth\relax
290   \box\FN@tempboxa
291 }
292 \AssignSocketPlug{fntext/make}{default}

```

`fntext/begin (socket)` By default adds a strut at the start of the footnote text.

```

293 \NewSocket      {fntext/begin}{0}
294 \NewSocketPlug{fntext/begin}{default}{ \rule\z@\footnotesep }
295 \AssignSocketPlug{fntext/begin}{default}

```

`fntext/end (socket)` By default adds a strut at the end of the footnote text unless we are no longer in hmode.

```

296 \NewSocket      {fntext/end}{0}
297 \NewSocketPlug{fntext/end}{default}{ \@finalstrut\strutbox }

```

When running several footnotes together as a paragraph some additional glue has to be added between them.

```

298 \NewSocketPlug{fntext/end}{para}
299 {%
300     \strut
301     \penalty-10\relax
302     \hskip\footglue
303 }
304 \AssignSocketPlug{fntext/end}{default}

```

tagsupport/fntext/begin (socket) Kernel sockets for tagging.

```
tagsupport/fntext/end (socket) 305 \NewSocket{tagsupport/fntext/begin}{0}
306 \NewSocket{tagsupport/fntext/end}{0}
```

Provide the name  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X} 2_{\epsilon}$  is used to and do this unconditionally (no patching of class code if any). This means that if a class provides it own definition that gets lost and if necessary needs to be handled with firstaid (or updating of the class).

```
307 \AddToHook{begindocument}
308 {
309   \cs_set_eq:NN \@footnotetext \fnote_footnotetext:n
310 }
```

## 5.12 The new \@makefntext command

`\footnotemargin` is the logic implemented by `footmisc`. Perhaps we don't want to do this like that in the kernel but for now I have used this interface unchanged.

```
311 \newdimen\footnotemargin
312 \footnotemargin\maxdimen          % no value given
313
314 \AtBeginDocument
315 {
316   \ifdim \footnotemargin=\maxdimen
317     \setlength\footnotemargin{1.8em}
318   \fi
319 }
```

`\fnote_makefntext:n`

```
320 \cs_new_protected:Npn \fnote_makefntext:n #1 {
```

Some classes in their redefinition for `\@makefntext` have placed some paragraph parameters at this point, but those can equally well go into the hook `fntext/para`. We therefore do not provide a further hook at this point.

```
321 \noindent
322 \socket_use:nn {tagsupport/fntext/mark} { \socket_use:n {fntext/mark} }
323 \socket_use:nn {tagsupport/fntext/text} { \socket_use:nn {fntext/text}{#1} }
324 }
```

*(End of definition for `\fnote_makefntext:n`. This function is documented on page ??.)*

`fntext/mark` (socket) A socket to typeset the mark at the start of a footnote.

```
325 \NewSocket {fntext/mark}{0}
```

The `default` plug implements the logic introduced with the `footmisc` package.

```
326 \NewSocketPlug{fntext/mark}{default}{
327   \ifdim\footnotemargin>\z@
328     \hb@xt@ \footnotemargin{\hss\@makefnmark}
329   \else
330     \ifdim\footnotemargin=\z@
331       \llap{\@makefnmark}
332     \else
333       \ifdim\footnotemargin=-\maxdimen
334         \@makefnmark
335       \else
336         \llap{\hb@xt@ -\footnotemargin{\@makefnmark\hss}}
```

```

337     \fi
338     \fi
339     \fi
340 }
341 \AssignSocketPlug{fntext/mark}{default}

```

`fntext/text` (*socket*) By default this socket does nothing special and simply processes its argument as provided.

```

342 \NewSocket    {fntext/text}{1}

```

`tagssupport/fntext/mark` (*socket*) Not a public socket but reserved for tagging. By default it contains `identity` and is `tagssupport/fntext/text` (*socket*) reassigned if tagging is active.

```

343 \NewSocket{tagssupport/fntext/mark}{1}
344 \NewSocket{tagssupport/fntext/text}{1}

```

### 5.12.1 Making documents use the new `\@makefntext`

If the definition for `\@makefntext` is that of the standard classes then replace it with `\fnote_makefntext:n`, otherwise try to patch the definition used in the class.

Here is the definition the way it is in `classes.dtx`. Notice that (for saving space) there is no space after `em` to terminate the assignment. We need to mimic that, otherwise a test would return false even if the definition has not been modified.

```

\old@std@class@makefntext

```

```

345 \newcommand\old@std@class@makefntext[1]{%
346     \parindent 1em%
347     \noindent
348     \hb@xt@1.8em{\hss\@makefnmark}#1}

```

*(End of definition for `\old@std@class@makefntext`. This function is documented on page ??.)*

Here is the messy code for patching. Note that this is only there to help classes along that aren't updated yet so it does some minimal patching to hopefully add kernel configuration hooks in the right place while otherwise leaving the legacy code alone. An updated class would not redefine `\@makefntext` but simply add appropriate code to the provided hooks.

What it does is roughly the following: It looks for a definition of `\@makefntext` of the form

```
{AAA \hbox BBB { CCC } DDD #1 EEE }
```

where “BBB” is something like `1em` or similar. It then replaces that with

```
{AAA \UseSocket{tagssupport/fntext/mark}{\hbox BBB { CCC }} DDD
\UseSocket{tagssupport/fntext/text}{#1} EEE }
```

The patching is not very careful, i.e., it assumes there is only one `#1` in the replacement text and that the first `\hbox` found is the right one to patch. But that is enough to cater for all definitions of `\@makefntext` out there in the TL distribution.

If `\hbox` is not found it tries the same looking for `\hb@xt@` which is what some classes use and if that is not found either it assume that this is a version that uses `\@makefnmark` without surrounding it in a box and if that fails it gives up with an `\ERROR` (which needs to get a proper definition).

```

349 \tl_new:N \l__fnote_patch_tl
350 \cs_new_eq:NN \__fnote_tmp:w \ERROR

```

```

351
352 \cs_new_protected:Npn \__fnote_patch:
353 {
354   \tl_set:No \l__fnote_patch_tl { \@makefnctext { \UseSocket{tagsupport/fntext/text}{##1} } }
355   \tl_if_in:NnTF \l__fnote_patch_tl { \hbox }
356     { \cs_set_eq:NN \__fnote_tmp:w \__fnote_patch_hbox:w }
357     {
358       \tl_if_in:NnTF \l__fnote_patch_tl { \hb@xt@ }
359         { \cs_set_eq:NN \__fnote_tmp:w \__fnote_patch_hb@xt@:w }
360         {

```

Some styles/classes use `\makebox[...][...]` instead of `\hb@xt@` so try to patch those too.

```

361       \tl_if_in:NnTF \l__fnote_patch_tl { \makebox }
362         { \cs_set_eq:NN \__fnote_tmp:w \__fnote_patch_makebox:w }
363         {
364           \tl_if_in:NnTF \l__fnote_patch_tl { \@makefnmark }
365             { \cs_set_eq:NN \__fnote_tmp:w \__fnote_patch_@makefnmark:w }
366             { \ERROR
367               \cs_set_eq:NN \__fnote_tmp:w \exp_stop_f: }
368           }
369       }
370   }
371   \tl_set:Nf \l__fnote_patch_tl
372     { \exp_after:wN \__fnote_tmp:w \l__fnote_patch_tl }
373   \cs_set:Npn \__fnote_tmp:w { \long \def \@makefnctext #####1 }
374   \exp_after:wN \__fnote_tmp:w \exp_after:wN { \l__fnote_patch_tl }
375 }

```

If `\@makefnctext` contains `\hbox` then grab “AAA” as #1 and “BBB” (up to the open `{}`) and return it as

```
AAA \@makefnctext@processX { \hbox BBB }
```

```

376 \cs_new:Npn \__fnote_patch_hbox:w #1 \hbox #2 #
377 { \exp_stop_f: #1 \@makefnctext@processX { \hbox #2 } }

```

Same for the other cases.

```

378 \cs_new:Npn \__fnote_patch_hb@xt@:w #1 \hb@xt@ #2 #
379 { \exp_stop_f: #1 \@makefnctext@processX { \hb@xt@ #2 } }
380 \cs_new:Npn \__fnote_patch_makebox:w #1 \makebox #2 #
381 { \exp_stop_f: #1 \@makefnctext@processX { \makebox #2 } }

```

If the definition contains neither `\hbox`, `\hb@xt@` nor `\makebox`, we see if it contains `\@makefnmark` and if so put the socket before that.

```

382 \cs_new:Npn \__fnote_patch_@makefnmark:w #1 \@makefnmark
383 { \exp_stop_f: #1 \@makefnctext@processX { \use:n } { \@makefnmark } }

```

The code provided by Bruno above expects 2 arguments but we need a different structure so this is a simple reshuffling. Would be better if we can patch the right structure in directly, but I’m not a patch person, so this is the simple way out for now:

```

384 \cs_new:Npn \@makefnctext@processX #1#2{\UseSocket{tagsupport/fntext/mark}{##1{##2}}}

```

At `\begin{document}` check if the current definition is that of the standard classes and if so replace it by `\fnote_makefntext:n` otherwise try and patch the definition made by some class or package using the approach above.

```

385
386 \AddToHook{begindocument}
387 {
388   \cs_if_eq:NNTF \@makefntext \old@std@class@makefntext
389     {
390       \cs_set_eq:NN \@makefntext \fnote_makefntext:n
391     }
392   {

```

If `\@makefntext` contains the definition from `footmisc` we do nothing, otherwise we try to patch.

```

393       \cs_if_eq:NNTF \@makefntext \footmisc@hang@makefntext
394         { \_fnote_patch: }
395     }
396   }
397
398
399 % possibly add the following to check for multiple \hbox in
400 % the definition:
401 %
402 % \seq_set_split:NnV \l__fnote_patch_seq { \hbox } \l__fnote_patch_tl
403 % \int_compare:nT { \seq_count:N \l__fnote_patch_seq } > 2 \ERROR
404 %

```

### 5.13 Document-level commands

`\footnotetext`

```

405 \DeclareDocumentCommand\footnotetext {o+m}
406 {
407   \fnote_set_fnmark:nn {#1} \@mpfn
408   \@footnotetext {#2}
409 }

```

*(End of definition for `\footnotetext`. This function is documented on page ??.)*

`\footnote`

```

410 \DeclareDocumentCommand\footnote {o+m}
411 {
412   \fnote_step_fnmark:nn {#1} \@mpfn
413   \cs_set_eq:NN \_fnote_footnotemark_finish: \prg_do_nothing:
414   \@footnotemark
415   \cs_set_eq:NN \_fnote_footnotemark_finish: \_fnote_footnotemark_default_finish:
416   \@footnotetext {#2}
417   \_fnote_footnotemark_finish:
418 }

```

*(End of definition for `\footnote`. This function is documented on page ??.)*

`\footnotemark`

```

419 \DeclareDocumentCommand\footnotemark {o}
420 {

```

```

421 \fnote_step_fmark:nn {#1} { footnote }
422 \@footnotemark
423 }

```

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

`\footref` `\footref` used the starred `\ref` in `\@thefnmark` as the linking is handled by the tagging code inside the `\@footnotemark`. `\footref` should not try to link to its related note automatically but should instead use the label. This is passed to `\@footnotemark` through `\l__fnote_currentlabel_tl`.

```

424 \DeclareDocumentCommand\footref {m}
425 {
426   \begingroup
427   \unrestored@protected@xdef\@thefnmark{\ref*{#1}}%
428   \endgroup
429   \bool_set_false:N \l__fnote_autodetect_bool
430   \tl_set:Nn \l__fnote_currentlabel_tl {#1}
431   \@footnotemark
432   \bool_set_true:N \l__fnote_autodetect_bool
433 }

```

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

## 5.14 Firstaid for packages and classes

### 5.15 Kernel patches

Tagging of footnotes in minipages require a change in the minipage commands We define at first a local configuration command for minipage footnotes.

```

434 \NewSocketPlug{fntext/process}{mp}
435 {
436   \global\setbox\@mpfootins\vbox{%
437     \unvbox\@mpfootins
438     #1
439   }
440 }

```

#### 5.15.1 memoir

The `memoir` class redefines various internal commands to inject its hooks and additional code. The following reinstates the kernel command and so probably breaks various options of `memoir`, but without the changes it errors anyway. The `footmisc` package should be used to change for example to para footnotes.

```

441 \AddToHook{class/memoir/before}
442 { \let\new@std@class@makecol\@makecol }
443 \AddToHook{class/memoir/after}
444 {
445   \cs_set_eq:NN \@footnotemark \fnote_footnotemark:
446   \cs_set_eq:NN \@makefntext\old@std@class@makefntext
447   \cs_set_eq:NN \@makecol\new@std@class@makecol
448 }

```



### 5.15.2 setspace

It should not overwrite it any longer but use a hook, so for now we do just that here.

```
449 \AddToHook{package/setspace/after}
450   {\let \@footnotetext \fnote_footnotetext:n
451     \AddToHook{fntext}[setspace]{\let\baselinestretch\setspace@singlespace}}
```

### 5.15.3 hyperref

hyperref has a hook which allows to disable its footnote related patches. As we will handle links directly in the code this is used.

```
452 \def\hyper@nopatch@footnote{}
```

We use the hyperref commands for now for links. To avoid to have to test for hyperref we provide dummies. TODO consider to use specials to get similar spacing.

```
453 \AtBeginDocument
454   {
455     \providecommand\hyper@linkstart{\@gobbletwo}
456     \providecommand\hyper@linkend{\@empty}
457   }
```

It must be possible to suppress the hyperlinking, both locally and globally. hyperref should set the boolean `\l_fnote_link_bool`. For now we test for the hyperref boolean (so it can be suppressed only globally).

```
458 \AtBeginDocument
459   {
460     \@ifpackageloaded{hyperref}
461       {
462         \legacy_if:nF{Hy@hyperfootnotes}{\bool_set_false:N \l_fnote_link_bool}
463       }
464       {
465         \bool_set_false:N \l_fnote_link_bool
466       }
467   }
```

## 5.16 Tagging and hyperlink code

### 5.16.1 Rolemap for structure tags

We use role-mapping to get more speaking names in the PDF and so ease debugging. These names are already provided by tagpdf directly.

### 5.16.2 Extending the label system

For `\footref` and (perhaps later for labeled footnotes) we must extend the label system. Beside the normal values we also need the structure number of the note. We use the inbuild label hook At first we define a suitable attribute, it uses as value the structure number of the note as stored in `\l__fnote_currentstruct_tl`

```
468 \property_new:nmmn {fnote/struct}{now}{1}{\l__fnote_currentstruct_tl}
```

We add a hook to the label hook. By default it does nothing

`\__fnote_label_hook:e`

```
469 \cs_new_protected:Npn \__fnote_label_hook:e #1 {}
470 \AddToHookWithArguments{label}{ \__fnote_label_hook:e{#1}}
```

(End of definition for `\__fnote_label_hook:e`.)

Inside a `footnotetext` we change the hook to store the structure number too. The name of label is provided as argument in the label hook.

```
471 \AddToHook{fntext/begin}
472 {
473   \cs_set_protected:Npn \__fnote_label_hook:e #1
474   {
475     \property_record:ee {\__fnote/#1} {fnote/struct}
476   }
477 }
```

### 5.16.3 Storing and retrieving reference data

To establish the connection between a mark and a note the mark has to store its representation, and the note has to analyse the stored representations to get the structure numbers of its mark. This is done with the public function to allow similar systems (e.g. tabular notes, other footnote series) to make use of this.

`\fnote_class_new:nn` This sets up a new footnote type, the first argument is the name, the second is meant for options. Currently it does nothing at all. It is not necessary to setup every footnote like command as its own type!

```
478 \cs_new_protected:Npn \fnote_class_new:nn #1 #2 % #1 name, #2 options
479 {
480   \prop_new:c { g__fnote_currentmarks_ #1 _prop }
481 }
482
483 \fnote_class_new:nn {default}{}
```

(End of definition for `\fnote_class_new:nn`. This function is documented on page 9.)

`\fnote_mark_gput:nn` This commands takes as argument the representation of the mark, e.g., `\@thefnmark` and the type (typically `default` should work).

```
484 \cs_new_protected:Npn \fnote_mark_gput:nn #1 #2 % #1 the representation of the mark, #2 type
485 {
486   \prop_gput:cen { g__fnote_currentmarks_ #2 _prop }
487   { \tag_get:n{struct_num} }
488   { #1 }
489 }
490
491 \cs_generate_variant:Nn \fnote_mark_gput:nn {no,oo}
```

(End of definition for `\fnote_mark_gput:nn`. This function is documented on page 9.)

`\fnote_mark_gpop_all:nnN` This commands takes as argument the representation of the mark (e.g. the content of `\@thefnmark`), the class (typically `default` should work) and a sequence into which every structure number in the property is stored that has the same value as the mark. The sequence is cleared first.

```
492 \cs_new_protected:Npn \fnote_mark_gpop_all:nnN #1 #2 #3
493 {
```

```

494 \seq_clear:N #3
495 \prop_set_eq:Nc \l__fnote_tmpa_prop { g__fnote_currentmarks_ #2 _prop }
496 \prop_map_inline:Nn \l__fnote_tmpa_prop
497 {
498   \tl_if_eq:nnT {#1} { ##2 }
499   {
store the key (the structure number) in the seq
500     \seq_put_right:Nn #3 { ##1 }
remove entry as used from the global prop
501     \prop_gremove:cn { g__fnote_currentmarks_ #2 _prop } {##1}
502   }
503 }
504 }
505 \cs_generate_variant:Nn\fnote_mark_gpop_all:nnN {ooN}

```

(End of definition for `\fnote_mark_gpop_all:nnN`. This function is documented on page 9.)

#### 5.16.4 Enabling tagging and links for the mark command

**FEMark** (*plug*) To handle the mark in the text, we define a special plug for the socket `tagsupport/fnmark` that receives `\@makefnstext` as its argument. At this time `\@thefnmark` is already set.

```

506 \NewSocketPlug{tagsupport/fnmark}{FEMark}
507 {

```

End an open mc and start the structure.

```

508   \tag_mc_end_push:
509   \tag_struct_begin:n { tag=footnotemark }

```

The associated note is either auto detected or given by the user.

```

510   \bool_if:NTF \l__fnote_autodetect_bool
511   {

```

For the auto detecting we store the structure number and `\@thefnmark` inside a prop and set the target name of the link to the current structure number. TODO: this should be usable for other footnote types which means the name of the prop shouldn't be fix.

```

512     \fnote_mark_gput:oo {\@thefnmark}{\l_fnote_type_tl}
513     \tl_set:Ne \l__fnote_linktarget_tl {footnote*.\tag_get:n{struct_num}}
514   }

```

If there is no autodetecting we need some id, currently it is called `\l__fnote_currentlabel_tl`. the Ref is set by looking at the label value. We must also add the current structure number to the `??f` the FEnote. Both must be delayed as we don't know if the objects of the FEnote and the mark have already been created.

```

515   {
516     \hook_gput_code:nne {tagpdf/finish/before} {tagpdf/footnote}
517     {
518       \exp_not:N\fnote_gput_refs:ee
519       { \tag_get:n{struct_num} }
520       { \property_ref:ee{ __fnote/\l__fnote_currentlabel_tl } {fnote/struct} }
521     }

```

in this case we set the name of the linktarget in the note to the structure number of the text mark.

```

522     \tl_set:Ne \l__fnote_linktarget_tl {footnote*.\property_ref:ee {__fnote/\l__fnote_cu
523   }

```

And now the actual content

```
524     \tag_mc_begin:n{tag=Lbl}
525     %
526     \bool_if:NTF \l_fnote_link_bool
527     {
528         \exp_args:No
529         \hyper@linkstart
530         { \l_fnote_link_type_tl }
531         { \l__fnote_linktarget_tl }
532         #1
533         \hyper@linkend
534     }
535     { #1 }
536     \tag_mc_end:
537     \tag_struct_end:
538     \tag_mc_begin_pop:n{}
539 }
```

At last assign the plug:

```
540 \AssignSocketPlug{tagsupport/fnmark}{FEMark}
```

### 5.16.5 The footnote text

We need a public command to append values to the Ref keys

```
\__fnote_gput_ref:nn
\fnote_gput_refs:nn
\fnote_gput_refs:ee
541 \cs_new_protected:Npn \__fnote_gput_ref:nn #1 #2 %#1 the structure number receiving the ref
542 {
543     \tag_if_active:T
544     {
545         \tag_struct_gput:nne {#1}{ref}{\tag_struct_object_ref:e { #2 }}
546     }
547 }
548 \cs_new_protected:Npn \fnote_gput_refs:nn #1 #2 % pair of numbers
549 {
550     \__fnote_gput_ref:nn {#1}{#2}
551     \__fnote_gput_ref:nn {#2}{#1}
552 }
553 \cs_generate_variant:Nn \fnote_gput_refs:nn {ee}
```

*(End of definition for \\_\_fnote\_gput\_ref:nn and \fnote\_gput\_refs:nn. This function is documented on page ??.)*

```
554 \tl_new:N \l__fnote_dflt_struct_tl
555 \tl_set:Nn \l__fnote_dflt_struct_tl {1}
```

kernel hooks for taggin this sets the structure around the whole text

**FENote** (*plug*)

```
556 \NewSocketPlug{tagsupport/fntext/begin}{FENote}
557 {
558     \tag_mc_end_push:
```

test if a footnote is allowed, if not move up to the document structure.

```

559   \tag_check_child:nnTF {FENote}{pdf2}
560   {
561     \tag_struct_begin:n { tag=footnote }
562   }
563   {
564     \tag_struct_begin:n { tag=footnote,parent=\l__fnote_dflt_struct_tl }
565   }

```

Store the current structure number for labels.

```

566   \tl_set:Ne \l__fnote_currentstruct_tl { \tag_get:n{struct_num} }

```

We want to move the label structure here. So we provide a container It number is `\l__fnote_currentstruct_tl` plus 1.

```

567   \tag_struct_begin:n { tag=footnotelabel }\tag_struct_end:

```

after we have opened the structure we can use the structure number to try to detect the connected marks. As with the marks we assume that sometimes no auto detection is done.

```

568   \bool_if:NTF \l__fnote_autodetect_bool
569   {

```

find open marks with identical `\@thefnmark`

```

570     \fnote_mark_gpop_all:ooN { \@thefnmark }{ \l_fnote_type_tl } \l__fnote_currentrefs_s

```

Then we store the object numbers of the marks in the `/Ref` of the `FENote` structure: and the number of the `FENote` into the marks structure:

```

571     \seq_map_inline:Nn \l__fnote_currentrefs_seq
572     {
573       \fnote_gput_refs:ee {##1}{ \l__fnote_currentstruct_tl }
574     }
575   }

```

If no auto detection is done

```

576     {%no auto
577
578     }
579   }

```

This finish the setup of the tagging structure.

Now we process the text. The destinations for the links are set with the label so that we can be sure that we are in hmode.

```

580 \NewSocketPlug{tagsupport/fntext/end}{FENote}
581 {
582   \tag_struct_end:
583   \tag_mc_begin_pop:n{
584   }

```

At last assign the plugs:

```

585 \AssignSocketPlug{tagsupport/fntext/begin}{FENote}
586 \AssignSocketPlug{tagsupport/fntext/end}{FENote}

```

The kernel socket `tagsupport/fntext/mark` is responsible for tagging the mark in the note. We use it to surround the mark with the needed tagging commands.

TODO check if additional kernel configuration points are needed. If yes, what about the paragraph start and the paratagging??

`FENoteLbl` (*plug*) This plug creates the label in the note on the bottom. It also adds link targets for the hyperlinking.

```

587 \NewSocketPlug{tagsupport/fntext/mark}{FENoteLbl}
588 {
589   \tag_mc_end_push:

```

We create a link target for every related mark. The name is `footnote*` (*structure number of the mark*). We also add a link target for the current structure (for `\footref`).

```

590   %\seq_show:N\l__fnote_currentrefs_seq
591   \seq_map_inline:Nn\l__fnote_currentrefs_seq {\MakeLinkTarget*{footnote*.\##1}}
592   \MakeLinkTarget*{footnote*.\l__fnote_currentstruct_tl}

```

Now we add the tagging commands. We move the structure of the label to to the container at the begin of the note.

```

593   \tag_struct_begin:n { tag=NonStruct,parent=\l__fnote_currentstruct_tl +1 }
594   \tag_mc_begin:n { tag=Lbl }
595   #1
596   \tag_mc_end:
597   \tag_struct_end:
598   \tag_mc_begin_pop:n{
599   }
600 \AssignSocketPlug{tagsupport/fntext/mark}{FENoteLbl}

```

`FENotetext` (*plug*) This plug is for the kernel socket `tagsupport/fntext/text` around the actual note text when doing tagging. Currently it only adds an MC chunk.

TODO Should it set a mc or could it rely on the content?

```

601 \NewSocketPlug{tagsupport/fntext/text}{FENotetext}
602 {
603   \tag_mc_end_push:
604   \tag_mc_begin:n{
605   #1
606   \tag_mc_end:
607   \tag_mc_begin_pop:n{
608   }
609 \AssignSocketPlug{tagsupport/fntext/text}{FENotetext}

```

---

```

610 \ExplSyntaxOff
611 </kernel>
612 <@@=)

```

## 6 Reimplementing the `footmisc` package

```

613 <{*footmisc}
614 %%
615 %% Copyright (c) 1995-2011 Robin Fairbairns
616 %% Copyright (c) 2018-2023 Robin Fairbairns, Frank Mittelbach
617 %%
618 %% This file is part of the 'latex-lab Bundle'.
619 %% -----

```

```

620 %%
621 %% It may be distributed and/or modified under the
622 %% conditions of the LaTeX Project Public License, either version 1.3c
623 %% of this license or (at your option) any later version.
624 %% The latest version of this license is in
625 %%   https://www.latex-project.org/lppl.txt
626 %% and version 1.3c or later is part of all distributions of LaTeX
627 %% version 2008 or later.
628 %%
629 %% This work has the LPPL maintenance status 'maintained'.
630 %%
631 \NeedsTeXFormat{LaTeX2e}
632 \providecommand\DeclareRelease[3]{}
633 \providecommand\DeclareCurrentRelease[2]{}
634
635 \DeclareRelease{v5}{2011-06-06}{footmisc-2011-06-06.sty}
636 \DeclareCurrentRelease{}{2022-02-14}
637 \ProvidesPackage{latex-lab-footmisc}%
638   [2022/03/08 v6.0d
639   a miscellany of footnote facilities -- latex-lab version%
640   ]
641
642 \NeedsTeXFormat{LaTeX2e}[2020/10/01]
643 \newtoks\FN@temptoken
644 \providecommand\protected@writeaux{%
645   \protected@write\@auxout
646 }
647 \def\l@advance@macro{\@@dvance@macro\edef}
648 \def\@@dvance@macro#1#2#3{\expandafter\@tempcnta#2\relax
649   \advance\@tempcnta#3\relax
650   #1#2{\the\@tempcnta}%
651 }
652 \let\l@advance@macro\l@advance@macro
653 \DeclareOption{symbol}{\renewcommand\thefootnote{\fnsymbol{footnote}}}
654 \newif\ifFN@robust \FN@robustfalse
655 \DeclareOption{symbol*}{%
656   \renewcommand\thefootnote{\@fnsymbol\c@footnote}%
657   \FN@robusttrue
658   \AtEndOfPackage{\setfnsymbol{lampost*-robust}}}%
659 }
660 \newif\ifFN@para \FN@parafalse
661 \DeclareOption{para}{%
Options are executed in the order of declaration, thus no point in checking for side option
as footmisc did in the past
662 %   \PackageError{footmisc}{Option "\CurrentOption" incompatible with
663 %                                     option "side"}%
664 %                                     {I shall ignore "\CurrentOption"}%
665   \FN@paratrue
666   \setlength\footnotemargin{-\maxdimen}   % default when para is used
667 }
668 \DeclareOption{side}{\ifFN@para
669   \PackageError{footmisc}{Option "\CurrentOption" incompatible with
670   option "para"}%

```

```

671             {I shall ignore "\CurrentOption"}%
672 \else
673   \AddToHook{fntext/para}{%
674     \hsize\marginparwidth      % correct the default \hsize
675     \footnotesep\z@           % don't add a default separation
676   }
677   \AssignSocketPlug{fntext/process}{side}
678 %   \AssignSocketPlug{fntext/make}{default}
679   \AssignSocketPlug{fntext/begin}{noop}
680   \AssignSocketPlug{fntext/end}{noop}
681 \fi
682 }
683 \let\footnotelayout\@empty
684 \DeclareOption{ragged}{%
685   \@ifundefined{RaggedRight}{%
686     {\renewcommand\footnotelayout{\linepenalty50 \raggedright}}%
687     {\renewcommand\footnotelayout{\linepenalty50 \RaggedRight}}%
688   }
689 \newif\ifFN@perpage
690 \FN@perpagefalse
691 \DeclareOption{perpage}{%
692   \FN@perpagetrue
693 }
694 \newif\ifFN@fixskip      \FN@fixskipfalse
695
696 \let\FN@bottomcases\thr@@
697 \newif\ifFN@abovefloats \FN@abovefloatstrue
698 \DeclareOption{bottom}{%
699   \let\FN@bottomcases\@ne
700   \FN@abovefloatsfalse
701   \FN@fixskiptrue
702 }
703 \DeclareOption{bottomfloats}{%
704   \let\FN@bottomcases\tw@
705   \FN@abovefloatstrue \FN@fixskiptrue
706 }
707 \DeclareOption{abovefloats}{\FN@abovefloatstrue \FN@fixskiptrue}
708 \DeclareOption{belowfloats}{\FN@abovefloatsfalse \FN@fixskiptrue}
709 \DeclareOption{marginal}{%
710   \footnotemargin-0.8em\relax
711 }
712 \DeclareOption{flushmargin}{%
713   \footnotemargin0pt\relax
714 }
715 \newif\ifFN@hangfoot \FN@hangfootfalse
716 \DeclareOption{hang}{%
717   \FN@hangfoottrue
718 }
719 \newcommand*\hangfootparskip{0.5\baselineskip}
720 \newcommand*\hangfootparindent{0em}%
721 \DeclareOption{norule}{%
722   \renewcommand\footnoterule{}%
723   \advance\skip\footins 4\p@\@plus2\p@\relax
724 }

```



```

725 \DeclareOption{splitrule}{%
726   \gdef\split@prev{0}
727   \let\pagefootnoterule\footnoterule
728   \let\mpfootnoterule\footnoterule
729   \def\splitfootnoterule{\kern-3\p@ \hrule \kern2.6\p@}
730   \def\footnoterule{\relax
731     \ifx \@listdepth\@mplistdepth
732       \mpfootnoterule
733     \else
734       \ifnum\split@prev=\z@
735         \pagefootnoterule
736       \else
737         \splitfootnoterule
738       \fi
739     \xdef\split@prev{\the\insertpenalties}%
740   \fi
741 }%
742 }
743 \newif\ifFN@stablefootnote \FN@stablefootnotefalse
744 \DeclareOption{stable}{\FN@stablefootnotetrue}
745 \newif\ifFN@multiplefootnote \FN@multiplefootnotefalse
746 \DeclareOption{multiple}{\FN@multiplefootnotetrue}
747 \ProcessOptions

```

Footnote box layout for para footnotes; this would also be the hook to support dbtfootnotes (from the dbtfoot package if we integrate that).

```

748 \ifFN@para
749   \NewSocketPlug{@makecol/footnotes}{para}{%
750     \global\setbox\footins\vbox{\FN@makefootnoteparagraph}%
751   }
752   \AssignSocketPlug{@makecol/footnotes}{para}
753 \fi
754 \ifFN@fixskip
755   \def\@outputbox@removebskip{%
756     \ifx\@textbottom\relax \else
757       \@outputbox@append{%
758         \@tempskipa\lastskip
759         \ifnum \gluestretchorder\@tempskipa>\z@
760           \vskip-\@tempskipa
761         \xdef\@outputbox@reinsertbskip
762           {\noexpand\@outputbox@append{\vskip\the\@tempskipa}}%
763       \else
764         \global\let\@outputbox@reinsertbskip\relax
765       \fi
766     }%
767   \fi
768 }
769 \let\@outputbox@reinsertbskip\relax
770 \else
771   \let\@outputbox@removebskip \relax
772   \let\@outputbox@reinsertbskip\relax
773 \fi
774 \ifcase \FN@bottomcases\relax
775   \ERROR

```

```

776 \or %1
777   \ifFN@abovefloats
778     \AssignSocketPlug {@makecol/outputbox}{space-footnotes-floats}
779   \else
780     \AssignSocketPlug {@makecol/outputbox}{floats-footnotes-space}
781   \fi
782 \or %2
783   \ifFN@abovefloats
784     \AssignSocketPlug {@makecol/outputbox}{footnotes-space-floats}
785   \else
786     \AssignSocketPlug {@makecol/outputbox}{space-floats-footnotes}
787   \fi
788 \or %3
789   \ifFN@abovefloats
790     \AssignSocketPlug {@makecol/outputbox}{footnotes-floats}
791   \else
792     \AssignSocketPlug {@makecol/outputbox}{floats-footnotes}
793   \fi
794 \else
795   \ERROR
796 \fi
797
798 % next can be dropped when cleaned up
799 \newif\ifFN@setspace
800 \@ifpackageloaded{setspace}%
801   {%
802     \FN@setspace>true
803     \@ifclassloaded{memoir}%
804       {%
805         \AddToHook{fntext}{\let\baselinestretch\m@m@singlespace}%
806         \let\FN@baselinestretch\m@m@singlespace
807       }%
808     {%
809       \AddToHook{fntext}{\let\baselinestretch\setspace@singlespace}%
810       \let\FN@baselinestretch\setspace@singlespace
811     }%
812   }%
813   {%
814     \FN@setspace>false
815   }
816
817
818
819 \ifFN@para
820
821 % \AssignSocketPlug{fntext/process}{default}
822 \AssignSocketPlug{fntext/make}{para}
823 \AssignSocketPlug{fntext/begin}{noop}
824 \AssignSocketPlug{fntext/end}{para}
825
826 \fi
827
828
829

```

```

830 \iffN@para
831   \let\FN@tempboxa\@tempboxa
832   \newbox\FN@tempboxb
833   \newbox\FN@tempboxc
834   \newskip\footglue \footglue=1em plus.3em minus.3em
835
836 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
837   \newdimen\footnotelineskip
838
839   % establish late:
840
841   \AddToHook{begindocument/before} {%
842     {%
843       \footnotesize
844       \global\footnotelineskip=\normalbaselineskip
845     }%
846   }

```

The coding is based on David Kastrup's improvement to Don Knuth's original implementation. You find in the  $\TeX$ book if you own the latest edition.

```

847
848   \long\def\FN@makefootnoteparagraph{%
849     \FN@setfootnoteparawidth
850     \@parboxrestore
851     \baselineskip=\footnotelineskip
852     \unvbox\footins \FN@removehboxes
853     \RawParEnd
854   }
855   \def\FN@removehboxes{\setbox\FN@tempboxa\lastbox
856     \ifhbox\FN@tempboxa{\FN@removehboxes}%
857     \unhbox\FN@tempboxa
858     \else
859       \RawNoindent
860       \rule\z@\footnotesep
861     \fi
862   }
863   \fi
864
865
866   \@ifpackageloaded{multicol}
867     {\def\FN@setfootnoteparawidth
868       {\hsize\ifnum\doublecol@number>\@ne
869         \textwidth
870         \else \columnwidth \fi}}
871     {\def\FN@setfootnoteparawidth{\hsize\columnwidth}}
872
873   \iffN@perpage
874     \RequirePackage{perpage}
875     \MakePerPage{footnote}

```

Fix a bug in perpage ...

```

876   \def\@stpelt#1{\global\csname c@#1\endcsname \m@ne
877     \stepcounter{#1}%
878     \pp@fix@MakePerPage{#1}%
879   }

```

```

880 \def\pp@fix@MakePerPage#1{%
881     \ifnum \value{#1}>\z@
882         \addtocounter{#1}\m@ne\fi
883 }

```

The above code may look a bit odd: the `\stepcounter` sets the counter to zero and then we alter it if it is not zero. The reason is that `\stepcounter` resets other counters and when `perpage` is loaded this results in updating counters on the reset list to 1 (or to a higher starting value if `\MakePerPage` is used with an optional argument, which is precisely the problem here). By subtracting 1 in that case we set it back to 1 lower than the starting value.

But to make this fully work we also need to update a support command in `perpage`:

```

884 \def\pp@c1@end@iii\stepcounter#1\pp@fix@MakePerPage#2{}
885 \fi
886
887
888 \ifFN@para
889
890 % This can use the default interface, except that a negative value for
891 % \footnotemargin makes little sense, so we test for this and warn if
892 % necessary. But -\maxdimen is ok again, so would need to be a litte bit more elaborate.
893 %
894
895 %\AddToHook{fntext/para}{
896 % \ifdim \footnotemargin >\z@ \else
897 % \PackageWarningNoline{footmisc}{Option 'para' needs positive \noexpand\footnotemargin}%
898 % \footnotemargin 1.8em\relax
899 % \fi
900 %}
901
902
903 \AddToHook{fntext/begin}{\nobreak \hspace{.2em}}
904
905 \else
906
907 \ifFN@hangfoot
908 \long\def\footmisc@hang@makefntext#1{%
909     \bgroup
910     \SuspendTagging{footmisc}%
911     \setbox\@tempboxa\hbox{%
912         \ifdim\footnotemargin>\z@
913             \hb@xt@\footnotemargin{\@makefnmark\hss}%
914         \else
915             \@makefnmark
916         \fi
917     }%
918     \leftmargin\wd\@tempboxa
919     \rightmargin\z@
920     \linewidth \columnwidth
921     \advance \linewidth -\leftmargin
922     \parshape \@ne \leftmargin \linewidth
923     \footnotesize
924     \@setpar{\@par}%
925     \ResumeTagging{footmisc}%

```

926           \leavevmode

Typesetting the mark twice means that one can't have any material inside that gets unhappy in that case. That shouldn't be a problem, but perhaps we have to come up with a more elaborate solution in the end.

```

927           \UseSocket{tagsupport/ftext/mark}%
928            {\llap{%
929              \ifdim\footnotemargin>\z@
930                \hb@xt@\footnotemargin{\@makefnmark\hss}%
931              \else
932                \@makefnmark
933              \fi
934            }}%
935            \parskip\hangfootparskip\relax
936            \parindent\hangfootparindent\relax
937            \footnotelayout#1%
938            \par
939            \egroup
940        }

```

Defined in a roundabout way so that we can test for it when patching classes that are not updated.

```

941        \let \@makefntext \footmisc@hang@makefntext
942
943        \else
944
945        % This is now using the default interface:
946        %
947        % \long\def\@makefntext#1{%
948        %    \parindent1em
949        %    \noindent
950        %    \ifdim\footnotemargin>\z@
951        %      \hb@xt@ \footnotemargin{\hss\@makefnmark}%
952        %    \else
953        %      \ifdim\footnotemargin=\z@
954        %        \llap{\@makefnmark}%
955        %      \else
956        %        \llap{\hb@xt@ -\footnotemargin{\@makefnmark\hss}}%
957        %      \fi
958        %    \fi
959        %    \footnotelayout#1%
960        % }
961
962        \fi
963        \fi
964
965
966
967
968        \ifFN@multiplefootnote
969        \providecommand*{\multiplefootnotemarker}{3sp}
970
971        We tag the separator as artifact
972        TODO: why is this done with \providecommand?
973
974        \ExplSyntaxOn

```

```

971 \providecommand*{\multfootsep}{\tag_mc_end_push:\tag_mc_begin:n{artifact},\tag_mc_end:\tag
972 \ExplSyntaxOff
973 \AddToHook{fnmark}      {\FN@mf@check}
974 \AddToHook{fnmark/end}  {\FN@mf@prepare}
975 %
976 \def\FN@mf@prepare{%
977   \kern-\multiplefootnotemarker
978   \kern\multiplefootnotemarker\relax
979 }
980 \def\FN@mf@check{%
981   \ifdim\lastkern=\multiplefootnotemarker\relax
982   ??? is that necessary or even correct ??
983   \edef\x@sf{\the\spacefactor}%
984   ??? shouldn't that be 2 unkerns ?? (none would also be ok)
985   \unkern % new
986   \unkern
987   \textsuperscript{\multfootsep}%
988   \spacefactor\x@sf\relax
989   \fi
990 }
991 \else
992   \let\FN@mf@prepare\relax
993   \fi
994 \ifFN@stablefootnote
995 \let\FN@sf@@footnote\footnote
996 \def\footnote{\ifx\protect\@typeset@protect
997   \expandafter\FN@sf@@footnote
998   \else
999   \expandafter\FN@sf@gobble@opt
1000  \fi
1001 }
1002 \edef\FN@sf@gobble@opt{\noexpand\protect
1003   \expandafter\noexpand\csname FN@sf@gobble@opt \endcsname}
1004 \expandafter\def\csname FN@sf@gobble@opt \endcsname{%
1005   \@ifnextchar[%]
1006     \FN@sf@gobble@twobrace
1007     \@gobble
1008 }
1009 \def\FN@sf@gobble@twobrace[#1]#2{}
1010 \let\FN@sf@@footnotemark\footnotemark
1011 \def\footnotemark{\ifx\protect\@typeset@protect
1012   \expandafter\FN@sf@@footnotemark
1013   \else
1014   \expandafter\FN@sf@gobble@optonly
1015   \fi
1016 }
1017 \edef\FN@sf@gobble@optonly{\noexpand\protect
1018   \expandafter\noexpand\csname FN@sf@gobble@optonly \endcsname}
1019 \expandafter\def\csname FN@sf@gobble@optonly \endcsname{%
1020   \@ifnextchar[%]
1021     \FN@sf@gobble@bracket
1022     {}%
1023 }
1024 \def\FN@sf@gobble@bracket[#1]{}

```

```

1025 \fi
1026 \newcommand\setfnsymbol[1]{%
1027   \@bsphack
1028   \@ifundefined{FN@fnsymbol@#1}{%
1029     {%
1030       \PackageError{footmisc}{Symbol style "#1" not known}%
1031       \@eha
1032     }{%
1033       \expandafter\let\expandafter\@fnsymbol\csname
1034         FN@fnsymbol@#1\endcsname
1035     }%
1036   \@esphack
1037 }
1038 \let\FN@fnsymbol@lamport\@fnsymbol
1039 \newif\if@tempswb
1040 \DeclareDocumentCommand\DefineFNSymbols {sm0{text}m}{-%
1041   \expandafter\ifx\csname FN@fnsymbol@#2\endcsname\relax
1042     \PackageInfo{footmisc}{Declaring symbol style #2}%
1043   \else
1044     \PackageWarning{footmisc}{Redeclaring symbol style #2}%
1045   \fi
1046   \toks@{}%
1047   \def\@tempb{\end}%
1048   \FN@build@symboldef#4\end
1049   \def\@tempc{math}%
1050   \def\@tempd{#3}%
1051   \expandafter\xdef\csname FN@fnsymbol@#2\endcsname##1{%
1052     \ifx\@tempc\@tempd
1053       \noexpand\ensuremath
1054     \else
1055       \noexpand\nfss@text
1056     \fi
1057     {%
1058       \noexpand\ifcase##1%
1059       \the\toks@
1060       \noexpand\else
1061       \IfBooleanTF#1{\noexpand\@ctrerr}%
1062         {\noexpand\FN@orange##1}%
1063       \noexpand\fi
1064     }%
1065   }%
1066 }
1067 \def\FN@build@symboldef#1{%
1068   \def\@tempa{#1}%
1069   \ifx\@tempa\@tempb
1070   \else
1071     \toks@\expandafter{\the\toks@\or#1}%
1072     \expandafter\FN@build@symboldef
1073   \fi
1074 }
1075 \DeclareDocumentCommand\DefineFNSymbolsTM {smm}{-%
1076   \expandafter\ifx\csname FN@fnsymbol@#2\endcsname\relax
1077     \PackageInfo{footmisc}{Declaring symbol style #2}%
1078   \else

```

```

1079     \PackageWarning{footmisc}{Redeclaring symbol style #2}%
1080     \fi
1081     \toks@{}%
1082     \def\@tempb{\end}%
1083     \FN@build@symboldefTM#3\end\@null
1084     \expandafter\xdef\csname FN@fnsymbol@#2\endcsname##1{%
1085         \noexpand\ifcase##1%
1086             \the\toks@
1087         \noexpand\else
1088             \IfBooleanTF#1{\noexpand\@ctrerr}%
1089                 {\noexpand\FN@orange##1}%
1090             \noexpand\fi
1091     }%
1092 }
1093 \def\FN@build@symboldefTM#1#2{%
1094     \def\@tempa{#1}%
1095     \ifx\@tempa\@tempb
1096     \else
1097         \toks@\expandafter{\the\toks@\or\TextOrMath{#1}{#2}}%
1098         \expandafter\FN@build@symboldefTM
1099     \fi
1100 }
1101 \def\FN@orange#1{%
1102     \ifFN@robust
1103         \@arabic#1%
1104         \@bsphack
1105         \PackageInfo{footmisc}{Footnote number \number#1 out of range}%
1106         \protect\@fnsymbol@orange
1107         \@esphack
1108     \else \@ctrerr \fi
1109 }
1110 \global\let\@diagnose@fnsymbol@orange\relax
1111 \AtEndDocument{\@diagnose@fnsymbol@orange}
1112 \def\@fnsymbol@orange{%
1113     \gdef\@diagnose@fnsymbol@orange{%
1114         \PackageWarningNoLine{footmisc}{Some footnote number(s)
1115             were out of range
1116             \MessageBreak
1117             see log for details%
1118         }%
1119     }%
1120 }
1121 \DefineFNSymbolsTM{bringhurst}{%
1122     \textasteriskcentered *%
1123     \textdagger \dagger
1124     \textdaggerdbl \ddagger
1125     \textsection \mathsection
1126     \textbardbl \||%
1127     \textparagraph \mathparagraph
1128 }%
1129 \DefineFNSymbolsTM{chicago}{%
1130     \textasteriskcentered *%
1131     \textdagger \dagger
1132     \textdaggerdbl \ddagger

```



```

1133 \textsection \mathsection
1134 \textbardbl \|%
1135 \#\#%
1136 }%
1137 \DefineFNsymbolsTM{wiley}{%
1138 \textasteriskcentered *%
1139 {\textasteriskcentered\textasteriskcentered}{**}%
1140 \textdagger \dagger
1141 \textdaggerdbl \ddagger
1142 \textsection \mathsection
1143 \textparagraph \mathparagraph
1144 \textbardbl \|%
1145 }%
1146 \DefineFNsymbolsTM{lamport-robust}{%
1147 \textasteriskcentered *%
1148 \textdagger \dagger
1149 \textdaggerdbl \ddagger
1150 \textsection \mathsection
1151 \textparagraph \mathparagraph
1152 \textbardbl \|%
1153 {\textasteriskcentered\textasteriskcentered}{**}%
1154 {\textdagger\textdagger}{\dagger\dagger}%
1155 {\textdaggerdbl\textdaggerdbl}{\ddagger\ddagger}%
1156 }
1157 \DefineFNsymbolsTM*{lamport*}{%
1158 \textasteriskcentered *%
1159 \textdagger \dagger
1160 \textdaggerdbl \ddagger
1161 \textsection \mathsection
1162 \textparagraph \mathparagraph
1163 \textbardbl \|%
1164 {\textasteriskcentered\textasteriskcentered}{**}%
1165 {\textdagger\textdagger}{\dagger\dagger}%
1166 {\textdaggerdbl\textdaggerdbl}{\ddagger\ddagger}%
1167 {\textsection\textsection}{\mathsection\mathsection}%
1168 {\textparagraph\textparagraph}{\mathparagraph\mathparagraph}%
1169 {\textasteriskcentered\textasteriskcentered\textasteriskcentered}{***}%
1170 {\textdagger\textdagger\textdagger}{\dagger\dagger\dagger}%
1171 {\textdaggerdbl\textdaggerdbl\textdaggerdbl}{\ddagger\ddagger\ddagger}%
1172 {\textsection\textsection\textsection}%%
1173 {\mathsection\mathsection\mathsection}%
1174 {\textparagraph\textparagraph\textparagraph}%%
1175 {\mathparagraph\mathparagraph\mathparagraph}%
1176 }
1177 \setfnsymbol{lamport*}
1178 \DefineFNsymbolsTM{lamport*-robust}{%
1179 \textasteriskcentered *%
1180 \textdagger \dagger
1181 \textdaggerdbl \ddagger
1182 \textsection \mathsection
1183 \textparagraph \mathparagraph
1184 \textbardbl \|%
1185 {\textasteriskcentered\textasteriskcentered}{**}%
1186 {\textdagger\textdagger}{\dagger\dagger}%

```

```

1187 {\textdaggerdbl\textdaggerdbl}{\ddagger\ddagger}%
1188 {\textsection\textsection}{\mathsection\mathsection}%
1189 {\textparagraph\textparagraph}{\mathparagraph\mathparagraph}%
1190 {\textasteriskcentered\textasteriskcentered\textasteriskcentered}{***}%
1191 {\textdagger\textdagger\textdagger}{\dagger\dagger\dagger}%
1192 {\textdaggerdbl\textdaggerdbl\textdaggerdbl}{\ddagger\ddagger\ddagger}%
1193 {\textsection\textsection\textsection}%%
1194   {\mathsection\mathsection\mathsection}%
1195 {\textparagraph\textparagraph\textparagraph}%%
1196   {\mathparagraph\mathparagraph\mathparagraph}%
1197 }
1198 \newcommand\mpfootnotemark{%
1199   \@ifnextchar[%]
1200     \@xmpfootnotemark
1201     {%
1202       \stepcounter\@mpfn
1203       \protected@xdef\@thefnmark{\thempfn}%
1204       \@footnotemark
1205     }%
1206 }
1207 \def\@xmpfootnotemark[#1]{%
1208   \begingroup
1209     \csname c@\@mpfn\endcsname #1\relax
1210     \unrestored@protected@xdef\@thefnmark{\thempfn}%
1211   \endgroup
1212   \@footnotemark
1213 }
1214 \endinput
1215 </footmisc>

```

## Index

The italic numbers denote the pages where the corresponding entry is described, numbers underlined point to the definition, all others indicate the places where it is used.

	<b>Symbols</b>		
\#	.....	<i>1135</i>	\AtEndDocument ..... <i>1111</i>
\	.....	<i>1126, 1134, 1144, 1152, 1163, 1184</i>	\AtEndOfPackage ..... <i>658</i>
	<b>A</b>		<b>B</b>
\addtocounter	.....	<i>882</i>	\baselineskip ..... <i>719, 851</i>
\AddToHook	.....		\baselinestretch ..... <i>451, 805, 809</i>
	. <i>307, 386, 441, 443, 449, 451, 471,</i>		\begingroup ..... <i>426, 1208</i>
	<i>673, 805, 809, 841, 895, 903, 973, 974</i>		\bgroup ..... <i>909</i>
\AddToHookWithArguments	.....	<i>470</i>	bool commands:
\advance	.....	<i>649, 723, 921</i>	\bool_gset_false:N ..... <i>118</i>
\AssignSocketPlug	....	<i>280, 292, 295,</i>	\bool_gset_true:N ..... <i>117</i>
	<i>304, 341, 540, 585, 586, 600, 609,</i>		\bool_if:NTF .. <i>121, 133, 510, 526, 568</i>
	<i>677, 678, 679, 680, 752, 778, 780,</i>		\bool_new:N ..... <i>69, 76, 116</i>
	<i>784, 786, 790, 792, 821, 822, 823, 824</i>		\bool_set_false:N ..... <i>429, 462, 465</i>
\AtBeginDocument	.....	<i>314, 453, 458</i>	\bool_set_true:N ..... <i>70, 77, 432</i>
			\box ..... <i>290</i>

<b>C</b>	
<code>\columnwidth</code> ..	16, 227, 289, 870, 871, 920
<code>\cs</code> .....	235, 236
cs commands:	
<code>\cs_generate_variant:Nn</code> .....	80, 81, 491, 505, 553
<code>\cs_gset_eq:NN</code> .....	128, 152
<code>\cs_if_eq:NNTF</code> .....	388, 393
<code>\cs_new:Npn</code> ...	376, 378, 380, 382, 384
<code>\cs_new_eq:NN</code> .....	197, 350
<code>\cs_new_protected:Npn</code> ...	82, 95, 117, 118, 119, 131, 155, 182, 200, 320, 352, 469, 478, 484, 492, 541, 548
<code>\cs_set:Npn</code> .....	373
<code>\cs_set_eq:NN</code> 199, 309, 356, 359, 362, 365, 367, 390, 413, 415, 445, 446, 447	
<code>\cs_set_protected:Npn</code> .....	473
<code>\csname</code> 20, 876, 1003, 1004, 1018, 1019, 1033, 1041, 1051, 1076, 1084, 1209	
<code>\CurrentOption</code> .....	662, 664, 669, 671
<b>D</b>	
<code>\dagger</code> ..	1123, 1131, 1140, 1148, 1154, 1159, 1165, 1170, 1180, 1186, 1191
<code>\ddagger</code> .	1124, 1132, 1141, 1149, 1155, 1160, 1166, 1171, 1181, 1187, 1192
<code>\DebugFNotesOff</code> .....	7, <u>117</u>
<code>\DebugFNotesOn</code> .....	7, <u>117</u>
<code>\DeclareCurrentRelease</code> .....	633, 636
<code>\DeclareDocumentCommand</code> .....	405, 410, 419, 424, 1040, 1075
<code>\DeclareOption</code> .....	653, 655, 661, 668, 684, 691, 698, 703, 707, 708, 709, 712, 716, 721, 725, 744, 746
<code>\DeclareRelease</code> .....	632, 635
<code>\def</code> 12, 18, 29, 30, 34, 230, 373, 452, 647, 648, 729, 730, 755, 848, 855, 867, 871, 876, 880, 884, 908, 947, 976, 980, 996, 1004, 1009, 1011, 1019, 1024, 1047, 1049, 1050, 1067, 1068, 1082, 1093, 1094, 1101, 1112, 1207	
default (plug) 5, 5, 5, 5, 5, 5, 5, 5, 5, 6, 20	
<code>\DefineFNsymbols</code> .....	1040
<code>\DefineFNsymbolsTM</code> .....	1075, 1121, 1129, 1137, 1146, 1157, 1178
<code>\dimexpr</code> .....	288
<code>\dp</code> .....	225, 286
<b>E</b>	
<code>\edef</code> .....	164, 647, 983, 1002, 1017
<code>\egroup</code> .....	939
<code>\else</code> .	329, 332, 335, 672, 733, 736, 756, 763, 770, 779, 785, 791, 794, 858, 870, 896, 905, 914, 931, 943, 952, 955, 991, 998, 1013, 1043, 1054, 1060, 1070, 1078, 1087, 1096, 1108
<code>\end</code> .....	1047, 1048, 1082, 1083
<code>\endcsname</code> .....	
..	20, 876, 1003, 1004, 1018, 1019, 1034, 1041, 1051, 1076, 1084, 1209
<code>\endgroup</code> .....	428, 1211
<code>\endinput</code> .....	1214
<code>\ensuremath</code> .....	1053
<code>\ERROR</code> .....	21, 350, 366, 403, 775, 795
exp commands:	
<code>\exp_after:wN</code> .....	372, 374
<code>\exp_args:No</code> .....	528
<code>\exp_not:N</code> .....	518
<code>\exp_stop_f:</code> ..	367, 377, 379, 381, 383
<code>\expandafter</code> .....	
..	648, 997, 999, 1003, 1004, 1012, 1014, 1018, 1019, 1033, 1041, 1051, 1071, 1072, 1076, 1084, 1097, 1098
<code>\ExplSyntaxOff</code> .....	610, 972
<code>\ExplSyntaxOn</code> .....	64, 970
<b>F</b>	
<code>FEMark</code> (plug) .....	<u>506</u>
<code>FENote</code> (plug) .....	<u>556</u>
<code>FENoteLbl</code> (plug) .....	<u>587</u>
<code>FENotetext</code> (plug) .....	<u>601</u>
<code>\fi</code> .....	173, 191, 318, 337, 338, 339, 681, 738, 740, 753, 765, 767, 773, 781, 787, 793, 796, 826, 861, 863, 870, 882, 885, 899, 916, 933, 957, 958, 962, 963, 989, 993, 1000, 1015, 1025, 1045, 1056, 1063, 1073, 1080, 1090, 1099, 1108
<code>\floatingpenalty</code> .....	226
<code>fnmark</code> (hook) .....	4, 4, <u>107</u>
<code>fnmark/after</code> (hook) ...	4, 4, 4, 6, 16, <u>107</u>
<code>fnmark/before</code> (hook) .....	4, 4, 4, <u>107</u>
<code>fnmark/begin</code> (hook) .....	4, 4, <u>107</u>
<code>fnmark/end</code> (hook) .....	4, 4, 16, <u>107</u>
fnote commands:	
<code>\fnote_class_new:nn</code> .	9, <u>478</u> , 478, 483
<code>\g_fnote_debug_bool</code> .....	116, 117, 118, 121, 133
<code>\fnote_footnotemark:</code> <u>155</u> , 155, 199, 445	
<code>\fnote_footnotetext:n</code> .....	<u>200</u> , 200, 309, 450
<code>\fnote_gput_refs:nn</code> .....	518, <u>541</u> , 548, 553, 573
<code>\l_fnote_link_bool</code> <u>25</u> , <u>69</u> , 462, 465, 526	
<code>\l_fnote_link_type_tl</code> .....	<u>71</u> , 530
<code>\fnote_makefntext:n</code> .....	10, 21, 23, <u>320</u> , 320, 390

\fnote_mark_gpop_all:nnN	.....	fntext/process (socket)	.....	5, 5, <u>277</u>
.....	9, <u>492</u> , 492, 505, <u>570</u>	fntext/text (socket)	.....	6, 6, <u>342</u>
\fnote_mark_gput:nn	.....	\footglue	.....	302, <u>834</u>
.....	9, <u>484</u> , 484, 491, <u>512</u>	\footins	.....	5, 278, 723, 750, 852
\fnote_set_fnmark:nn	....	\footnote	.....	5, 8, <u>10</u> , <u>19</u> , <u>410</u> , 995, 996
\fnote_step_fnmark:nn	<u>82</u> , 82, <u>412</u> , <u>421</u>	\footnotebaselineskip	..	289, 837, 844, 851
\l_fnote_type_tl	.....	\footnotelayout	..	683, 686, 687, 937, 959
fnote internal commands:		\footnotemargin	....	5, 7, <u>20</u> , 311, 312, 316, 317, 327, 328, 330, 333, 336, 666, 710, 713, 891, 896, 897, 898, 912, 913, 929, 930, 950, 951, 953, 956
\l__fnote_autodetect_bool	.....	\footnotemark	...	4, 8-10, <u>419</u> , 1010, 1011
.....	<u>76</u> , 429, 432, 510, <u>568</u>	\footnoterule	.....	722, 727, 728, 730
\l__fnote_currentlabel_tl	.....	\footnotesep	.....	23, 224, 294, 675, 860
.....	<u>24</u> , <u>27</u> , <u>74</u> , 430, 520, <u>522</u>	\footnotesize	.....	15, 217, 843, 923
\l__fnote_currentrefs_seq	.....	\footnotetext	.....	5, 6, 8-10, <u>19</u> , <u>405</u>
.....	<u>75</u> , 570, 571, 590, 591	\footref	.....	9, <u>10</u> , <u>24</u> , <u>25</u> , <u>30</u> , <u>424</u>
\l__fnote_currentstruct_tl	.....			
...	<u>25</u> , 78, 79, 468, 566, 573, <u>592</u> , <u>593</u>	<b>G</b>		
\__fnote_debug_footnotemark:	...	\gdef	.....	726, 1113
.....	<u>119</u> , <u>128</u> , <u>156</u>	\global	..	13, 436, 750, 764, 844, 876, 1110
\__fnote_debug_footnotetext:	...	\gluestretchorder	.....	759
.....	<u>131</u> , <u>152</u> , <u>201</u>	group commands:		
\l__fnote_dflt_struct_tl	<u>554</u> , <u>555</u> , <u>564</u>	\group_begin:	.....	89, 101
\__fnote_footmark_finish:	.....	\group_end:	.....	92, 104
.....	<u>16</u>			
\__fnote_footnotemark_default_finish:	.....	<b>H</b>		
.....	<u>16</u> , <u>182</u> , 182, 197, <u>415</u>	\hangfootparindent	.....	720, 936
\__fnote_footnotemark_finish:	...	\hangfootparskip	.....	719, 935
.....	<u>16</u> , 180, <u>182</u> , 197, 413, 415, 417	\hbox	.....	<u>21</u> , 22, 34, 285, 355, 376, 377, 399, 402, 911
\__fnote_gput_ref:nn	<u>541</u> , 541, 550, <u>551</u>	hook commands:		
\__fnote_label_hook:n	.....	\hook_gput_code:nnn	.....	80, 516
.....	<u>469</u> , 469, 470, <u>473</u>	\hook_log:n	.....	123, 124, 125, 126, 127, 146, 147, 148, 149, 150, 151
\l__fnote_linktarget_tl	.....	\hook_use:n	.....	160, 170, 176, 204, 214, 239, 256, 263, 274
.....	<u>73</u> , 513, 522, <u>531</u>	Hooks:		
\__fnote_patch:	.....	fnmark	.....	4, 4, <u>107</u>
.....	<u>352</u> , <u>394</u>	fnmark/after	.....	4, 4, 4, 6, <u>16</u> , <u>107</u>
\__fnote_patch_@makefnmark:w	<u>365</u> , <u>382</u>	fnmark/before	.....	4, 4, 4, <u>107</u>
\__fnote_patch_hb@xt@:w	....	fnmark/begin	.....	4, 4, <u>107</u>
.....	359, 378	fnmark/end	.....	4, 4, <u>16</u> , <u>107</u>
\__fnote_patch_hbox:w	....	fntext	.....	6, 6, <u>111</u>
.....	356, 376	fntext/after	.....	6, 6, 6, <u>111</u>
\__fnote_patch_makebox:w	...	fntext/before	.....	6, 6, 6, <u>111</u>
.....	362, 380	fntext/begin	.....	6, 6, 6, <u>111</u>
\l__fnote_patch_seq	.....	fntext/end	.....	6, 6, 6, <u>111</u>
.....	402, 403	fntext/para	.....	6, 6, 6, 6, <u>20</u> , <u>111</u>
\l__fnote_patch_tl	.....	\hrule	.....	729
.....	349, 354, 355, 358, 361, 364, 371, 372, 374, 402	\hsize	.....	6, 18, 16, 227, 674, 868, 871
\__fnote_tmp:w	.....	\hskip	.....	302
.....	350, 356, 359, 362, 365, 367, 372, 373, 374	\hspace	.....	903
\l__fnote_tmpa_prop	....	\hss	.....	328, 336, 348, 913, 930, 951, 956
.....	65, 495, 496	\ht	.....	287
\l__fnote_tmpa_tl	.....			
.....	66			
\fnsymbol	.....			
.....	653			
fntext (hook)	.....			
.....	6, 6, <u>111</u>			
fntext/after (hook)	.....			
.....	6, 6, 6, <u>111</u>			
fntext/before (hook)	.....			
.....	6, 6, 6, <u>111</u>			
fntext/begin (hook)	.....			
.....	6, 6, 6, <u>111</u>			
fntext/begin (socket)	.....			
.....	5, 5, <u>293</u>			
fntext/end (hook)	.....			
.....	6, 6, 6, <u>111</u>			
fntext/end (socket)	.....			
.....	5, 5, <u>296</u>			
fntext/make (socket)	..			
.....	5, 5, 5, 5, 5, 6, <u>281</u>			
fntext/mark (socket)	.....			
.....	5, 5, <u>325</u>			
fntext/para (hook)	....			
.....	6, 6, 6, 6, <u>20</u> , <u>111</u>			

<b>I</b>	
identity (plug) . . . . .	6, 16, 17, 21
\IfBooleanTF . . . . .	1061, 1088
\ifcase . . . . .	774, 1058, 1085
\ifdim . . . . .	316, 327,
	330, 333, 896, 912, 929, 950, 953, 981
\ifhbox . . . . .	856
\ifhmode . . . . .	163, 189
\ifnum . . . . .	734, 759, 868, 881
\ifx . . . . .	731, 756,
	996, 1011, 1041, 1052, 1069, 1076, 1095
\ignorespaces . . . . .	23, 258
\insert . . . . .	5, 16, 278
\insertpenalties . . . . .	739
int commands:	
\int_compare:nTF . . . . .	403
\int_set:Nn . . . . .	90, 102
\interfootnotelinepenalty . . . . .	223
\interlinepenalty . . . . .	6, 223
<b>K</b>	
\kern . . . . .	729, 977, 978
<b>L</b>	
\label . . . . .	9
\lastbox . . . . .	855
\lastkern . . . . .	981
\lastskip . . . . .	758
\leavevmode . . . . .	162, 926
\leftmargin . . . . .	918, 921, 922
legacy commands:	
\legacy_if:nTF . . . . .	462
\let . . . . .	442, 450, 451, 652, 683,
	696, 699, 704, 727, 728, 764, 769,
	771, 772, 805, 806, 809, 810, 831,
	941, 992, 995, 1010, 1033, 1038, 1110
\linepenalty . . . . .	686, 687
\linewidth . . . . .	920, 921, 922
\llap . . . . .	5, 7, 331, 336, 928, 954, 956
\long . . . . .	12, 373, 848, 908, 947
\ltxlabfootnotedate . . . . .	4
\ltxlabfootnoteversion . . . . .	4
<b>M</b>	
\makebox . . . . .	22, 361, 380, 381
\MakeLinkTarget . . . . .	591, 592
\MakePerPage . . . . .	36, 875
\marginpar . . . . .	279
\marginparwidth . . . . .	674
\mathparagraph . . . . .	1127, 1143, 1151,
	1162, 1168, 1175, 1183, 1189, 1196
\mathsection . . . . .	1125, 1133, 1142, 1150,
	1161, 1167, 1173, 1182, 1188, 1194
\maxdimen . . . . .	5, 7, 312, 316, 333, 666, 892
\MessageBreak . . . . .	1116
mp (plug) . . . . .	5
\mpfootnotemark . . . . .	1198
\mpfootnoterule . . . . .	728, 732
\multfootsep . . . . .	971, 987
\multiplefootnotemarker . . . . .	969, 977, 978, 981
<b>N</b>	
\NeedsTeXFormat . . . . .	631, 642
\newbox . . . . .	832, 833
\newcommand . . . . .	345, 719, 720, 1026, 1198
\newdimen . . . . .	311, 837
\NewHook . . . . .	108, 109, 110, 112, 113, 114, 115
\newif . . . . .	654, 660,
	689, 694, 697, 715, 743, 745, 799, 1039
\NewMirroredHookPair . . . . .	107, 111
\newskip . . . . .	834
\NewSocket . . . . .	198, 277, 281,
	293, 296, 305, 306, 325, 342, 343, 344
\NewSocketPlug . . . . .	
	278, 279, 282, 283, 294, 297, 298,
	326, 434, 506, 556, 580, 587, 601, 749
\newtoks . . . . .	643
\nobreak . . . . .	4, 172, 903
\noexpand . . . . .	
	762, 897, 1002, 1003, 1017, 1018,
	1053, 1055, 1058, 1060, 1061, 1062,
	1063, 1085, 1087, 1088, 1089, 1090
\noindent . . . . .	321, 347, 949
noop (plug) . . . . .	5, 5, 5, 6
\normalbaselineskip . . . . .	844
\normalfont . . . . .	34
\number . . . . .	1105
<b>O</b>	
\or . . . . .	776, 782, 788, 1071, 1097
<b>P</b>	
\PackageError . . . . .	662, 669, 1030
\PackageInfo . . . . .	1042, 1077, 1105
\PackageWarning . . . . .	1044, 1079
\PackageWarningNoLine . . . . .	1114
\PackageWarningNoline . . . . .	897
\pagefootnoterule . . . . .	727, 735
\par . . . . .	24, 267, 938
para (plug) . . . . .	5, 5, 5
\parindent . . . . .	6, 229, 346, 936, 948
\parshape . . . . .	922
\parskip . . . . .	935
\penalty . . . . .	301
Plugs:	
default . . . . .	5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 6, 20
FEMark . . . . .	506
FENote . . . . .	556

FENoteLbl	587
FENotetext	601
identity	6, 16, 17, 21
mp	5
noop	5, 5, 5, 6
para	5, 5, 5
side	5
prg commands:	
\prg_do_nothing:	128, 152, 413
\ProcessOptions	747
prop commands:	
\prop_gput:Nnn	486
\prop_gremove:Nn	501
\prop_map_inline:Nn	496
\prop_new:N	65, 480
\prop_set_eq:NN	495
property commands:	
\property_new:nnnn	468
\property_record:nn	475
\property_ref:nn	520, 522
\protect	996, 1002, 1011, 1017, 1106
\providecommand	37, 455, 456, 632, 633, 644, 969, 971
\ProvidesFile	3
\ProvidesPackage	637
R	
\RaggedRight	687
\raggedright	686
\RawNoindent	859
\RawParEnd	853
\ref	24, 427
\relax	190, 289, 301, 648, 649, 710, 713, 723, 730, 756, 764, 769, 771, 772, 774, 898, 935, 936, 978, 981, 988, 992, 1041, 1076, 1110, 1209
\renewcommand	653, 656, 686, 687, 722
\RequirePackage	874
\ResumeTagging	925
\rightmargin	919
\rule	23, 294, 860
S	
seq commands:	
\seq_clear:N	494
\seq_count:N	403
\seq_map_inline:Nn	571, 591
\seq_new:N	75
\seq_put_right:Nn	500
\seq_set_split:Nnn	402
\seq_show:N	590
\setbox	13, 285, 436, 750, 855, 911
\setsymbol	658, 1026, 1177
\setlength	317, 666
side (plug)	5
\skip	723
socket commands:	
\socket_log:n	135, 136, 137, 138, 139, 140, 141, 142, 143, 144, 145
\socket_use:n	206, 252, 265, 271, 322
\socket_use:nn	178, 207, 247, 322, 323
Sockets:	
fntext/begin	5, 5, 293
fntext/end	5, 5, 296
fntext/make	5, 5, 5, 5, 5, 6, 281
fntext/mark	5, 5, 325
fntext/process	5, 5, 277
fntext/text	6, 6, 342
tagssupport/fnmark	7, 7, 10, 27, 198
tagssupport/fntext/begin	8, 8, 10, 305
tagssupport/fntext/end	8, 8, 10, 305
tagssupport/fntext/mark	8, 8, 10, 29, 343
tagssupport/fntext/text	8, 8, 10, 30, 343
\space	4
\spacefactor	164, 190, 983, 988
\splitfootnoterule	729, 737
\splitmaxdepth	225
\splittopskip	224
\stepcounter	36, 85, 877, 884, 1202
\strut	300
\strutbox	23, 225, 297
\SuspendTagging	910
T	
tag commands:	
\tag_check_child:nnTF	559
\tag_get:n	487, 513, 519, 566
\tag_if_active:TF	543
\tag_mc_begin:n	524, 594, 604, 971
\tag_mc_begin_pop:n	538, 583, 598, 607, 971
\tag_mc_end:	536, 596, 606, 971
\tag_mc_end_push:	508, 558, 589, 603, 971
\tag_struct_begin:n	509, 561, 564, 567, 593
\tag_struct_end:	537, 567, 582, 597
\tag_struct_gput:nnn	545
\tag_struct_object_ref:n	545
\tag_struct_use:n	81
tagssupport/fnmark (socket)	7, 7, 10, 27, 198
tagssupport/fntext/begin (socket)	8, 8, 10, 305
tagssupport/fntext/end (socket)	8, 8, 10, 305
tagssupport/fntext/mark (socket)	8, 8, 10, 29, 343
tagssupport/fntext/text (socket)	8, 8, 10, 30, 343

T<sub>E</sub>X and L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> commands:

<code>\@@dvance@macro</code>	647, 648	<code>\@tempb</code>	1047, 1069, 1082, 1095
<code>\@par</code>	924	<code>\@tempboxa</code>	831, 911, 918
<code>\@MM</code>	226	<code>\@tempc</code>	1049, 1052
<code>\@advance@macro</code>	652	<code>\@tempcnta</code>	648, 649, 650
<code>\@arabic</code>	1103	<code>\@tempd</code>	1050, 1052
<code>\@auxout</code>	645	<code>\@tempskipa</code>	758, 759, 760, 762
<code>\@bsphack</code>	1027, 1104	<code>\@textbottom</code>	756
<code>\@ctrerr</code>	1061, 1088, 1108	<code>\@textsuperscript</code>	34
<code>\@currentcounter</code>	18, 230	<code>\@thefnmark</code>	8–10, 12, 14, 24, 26, 27, 29, 20, 34, 86, 91, 98, 103, 231, 427, 512, 570, 1203, 1210
<code>\@currentlabel</code>	19, 231	<code>\@typeset@protect</code>	996, 1011
<code>\@diagnose@fnsymbol@orange</code>	1110, 1111, 1113	<code>\@x@sf</code>	164, 190, 983, 988
<code>\@eha</code>	1031	<code>\@xmpfootnotemark</code>	1200, 1207
<code>\@empty</code>	456, 683	<code>\@c@footnote</code>	656
<code>\@esphack</code>	1036, 1107	<code>\@color@begingroup</code>	21, 241
<code>\@finalstrut</code>	23, 297	<code>\@color@endgroup</code>	25, 268
<code>\@fnsymbol</code>	656, 1033, 1038	<code>\@doublecol@number</code>	868
<code>\@fnsymbol@orange</code>	1106, 1112	<code>\@FN@abovefloatsfalse</code>	700, 708
<code>\@footnotemark</code>	3, 16, 24, 199, 414, 422, 431, 445, 1204, 1212	<code>\@FN@abovefloatstrue</code>	697, 705, 707
<code>\@footnotetext</code>	3, 17, 309, 408, 416, 450	<code>\@FN@baselinestretch</code>	806, 810
<code>\@gobble</code>	1007	<code>\@FN@bottomcases</code>	696, 699, 704, 774
<code>\@gobbletwo</code>	455	<code>\@FN@build@symboldef</code>	1048, 1067, 1072
<code>\@ifclassloaded</code>	803	<code>\@FN@build@symboldefTM</code>	1083, 1093, 1098
<code>\@ifnextchar</code>	1005, 1020, 1199	<code>\@FN@fixskipfalse</code>	694
<code>\@ifpackageloaded</code>	460, 800, 866	<code>\@FN@fixskiptrue</code>	701, 705, 707, 708
<code>\@ifundefined</code>	685, 1028	<code>\@FN@fnsymbol@lampport</code>	1038
<code>\@listdepth</code>	731	<code>\@FN@hangfootfalse</code>	715
<code>\@makecol</code>	442, 447	<code>\@FN@hangfoottrue</code>	717
<code>\@makefnmark</code>	3–5, 7, 10, 16, 21, 22, 34, 178, 328, 331, 334, 336, 348, 364, 382, 383, 913, 915, 930, 932, 951, 954, 956	<code>\@FN@makefootnoteparagraph</code>	750, 848
<code>\@makefntext</code>	3, 5, 6, 10, 11, 20–23, 27, 22, 229, 243, 282, 285, 354, 373, 388, 390, 393, 446, 941, 947	<code>\@FN@m@f@check</code>	973, 980
<code>\@makefntext@processX</code>	377, 379, 381, 383, 384	<code>\@FN@m@f@prepare</code>	974, 976, 992
<code>\@mpfn</code>	29, 407, 412, 1202, 1209	<code>\@FN@multiplefootnotefalse</code>	745
<code>\@mpfootins</code>	13, 14, 436, 437	<code>\@FN@multiplefootnotetrue</code>	746
<code>\@mpfootnotetext</code>	9, 12	<code>\@FN@orange</code>	1062, 1089, 1101
<code>\@mplistdepth</code>	731	<code>\@FN@parafalse</code>	660
<code>\@ne</code>	699, 868, 922	<code>\@FN@paratrue</code>	665
<code>\@null</code>	1083	<code>\@FN@perpagefalse</code>	690
<code>\@outputbox@append</code>	757, 762	<code>\@FN@perpagetrue</code>	692
<code>\@outputbox@reinsertbskip</code>	761, 764, 769, 772	<code>\@FN@removehboxes</code>	852, 855, 856
<code>\@outputbox@removebskip</code>	755, 771	<code>\@FN@robustfalse</code>	654
<code>\@parboxrestore</code>	17, 228, 850	<code>\@FN@robusttrue</code>	657
<code>\@plus</code>	723	<code>\@FN@setfootnoteparawidth</code>	849, 867, 871
<code>\@setpar</code>	924	<code>\@FN@setspacefalse</code>	814
<code>\@stpelt</code>	876	<code>\@FN@setspacetrue</code>	802
<code>\@tempa</code>	1068, 1069, 1094, 1095	<code>\@FN@sf@@@footnote</code>	995, 997
		<code>\@FN@sf@@@footnotemark</code>	1010, 1012
		<code>\@FN@sf@gobble@bracket</code>	1021, 1024
		<code>\@FN@sf@gobble@opt</code>	999, 1002
		<code>\@FN@sf@gobble@optonly</code>	1014, 1017
		<code>\@FN@sf@gobble@twobracket</code>	1006, 1009
		<code>\@FN@stablefootnotefalse</code>	743
		<code>\@FN@stablefootnotetrue</code>	744

\FN@tempboxa .....	285,	\textasteriskcentered .....	1122, 1130, 1138, 1139, 1147, 1153,
286, 287, 288, 290, 831, 855, 856, 857		1158, 1164, 1169, 1179, 1185, 1190	
\FN@tempboxb .....	832	\textbardbl .....	1126, 1134, 1144, 1152, 1163, 1184
\FN@tempboxc .....	833	\textdagger .....	1123, 1131, 1140, 1148, 1154,
\FN@temptoken .....	643	1159, 1165, 1170, 1180, 1186, 1191	
\footmisc@hang@makefntext .....	393, 908, 941	\textdaggerdbl .....	1124, 1132, 1141, 1149, 1155,
\hb@xt@ .....	21, 22, 328, 336,	1160, 1166, 1171, 1181, 1187, 1192	
348, 358, 378, 379, 913, 930, 951, 956		\TextOrMath .....	1097
\hyper@linkend .....	456, 533	\textparagraph .....	1127, 1143, 1151,
\hyper@linkstart .....	455, 529	1162, 1168, 1174, 1183, 1189, 1195	
\hyper@nopatch@footnote .....	452	\textsection .....	1125, 1133, 1142, 1150,
\if@tempswb .....	1039	1161, 1167, 1172, 1182, 1188, 1193	
\ifFN@abovefloats .....	697, 777, 783, 789	\textsuperscript .....	987
\ifFN@fixskip .....	694, 754	\textwidth .....	869
\ifFN@hangfoot .....	715, 907	\the .....	164, 650,
\ifFN@multiplefootnote .....	745, 968	739, 762, 983, 1059, 1071, 1086, 1097	
\ifFN@para .....	660, 668, 748, 819, 830, 888	\tfootnote .....	30, 653, 656
\ifFN@perpage .....	689, 873	\thempfn .....	30, 1203, 1210
\ifFN@robust .....	654, 1102	tl commands:	
\ifFN@setspace .....	799	\tl_if_eq:nTF .....	498
\ifFN@stablefootnote .....	743, 994	\tl_if_in:NnTF .....	355, 358, 361, 364
\l@advance@macro .....	647, 652	\tl_if_novalue:nTF .....	83, 96
\m@m@singlespace .....	805, 806	\tl_new:N .....	66, 67, 71, 73, 74, 78, 349, 554
\m@ne .....	876, 882	\tl_set:Nn .....	68,
\new@std@class@makecol .....	442, 447	72, 79, 354, 371, 430, 513, 522, 555, 566	
\nfss@text .....	1055		
\old@std@class@makefntext .....	345, 388, 446		
\p@ .....	723, 729		
\p@footnote .....	231		
\pp@cl@end@iii .....	884		
\pp@fix@MakePerPage .....	878, 880, 884		
\protected@edef .....	19, 231		
\protected@write .....	645		
\protected@writeaux .....	644		
\protected@xdef .....	86, 98, 1203		
\reset@font .....	15, 216		
\setspace@singlespace .....	451, 809, 810		
\split@prev .....	726, 734, 739		
\thr@@ .....	696		
\toks@ .....	1046, 1059, 1071, 1081, 1086, 1097		
\tw@ .....	704		
\unrestored@protected@xdef .....	91, 103, 427, 1210		
\z@ .....	23,		
286, 294, 327, 330, 675, 734, 759,			
860, 881, 896, 912, 919, 929, 950, 953			

  

<b>U</b>	
\unhbox .....	857
\unkern .....	985, 986
\unskip .....	159
\unvbox .....	14, 437, 852
use commands:	
\use:N .....	86, 91, 98, 103
\use:n .....	383
\UseHook .....	187, 194
\UseSocket .....	354, 384, 927
<b>V</b>	
\value .....	881
\vbox .....	13, 436, 750
\vskip .....	760, 762
<b>W</b>	
\wd .....	288, 918
<b>X</b>	
\xdef .....	739, 761, 1051, 1084