

The `latex-lab-footnotes` code^{*}

Frank Mittelbach, L^AT_EX Project

June 1, 2025

Abstract

to be written

Contents

1	Introduction	2
1.1	Configuration methods	3
2	Sockets and hooks	3
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
3	Tagging and hyperlinking support	7
3.1	Technical details for the tagging	7
3.2	Requirements for links	9
3.3	The algorithmus to connect marks and notes	9
3.3.1	<code>\footref</code>	10
3.3.2	<code>\footnotemark</code> after <code>\footnotetext</code>	10
3.4	Links	10
3.5	Implementation details regarding tagging	10
3.6	Handling the mark	10
3.7	Handling the footnotetext	11
3.8	Footnotes in minipages	11
4	TODOs	11

*

5	The Implementation	11
5.1	File declaration	11
5.2	code not fully handled yet	11
5.3	Temporary variables	13
5.4	Public variables	13
5.5	Internal variables	13
5.6	Variants	14
5.7	Updating <code>\@thefnmark</code>	14
5.8	Hooks	15
5.9	Debugging code	15
5.10	The new <code>\@footnotemark</code> command	16
5.11	The new <code>\@footnotetext</code> command	18
5.12	The new <code>\@makefnmark</code> command	21
5.12.1	Making documents use the new <code>\@makefnmark</code>	22
5.13	Document-level commands	25
5.14	Firstaid for packages and classes	26
5.15	Kernel patches	26
5.15.1	<code>memoir</code>	26
5.15.2	<code>setspace</code>	26
5.15.3	<code>hyperref</code>	26
5.16	Tagging and hyperlink code	27
5.16.1	Rolemap for structure tags	27
5.16.2	Extending the label system	27
5.16.3	Storing and retrieving reference data	28
5.16.4	Enabling tagging and links for the mark command	29
5.16.5	The footnote text	30
6	Reimplementing the <code>footmisc</code> package	33
	Index	45

1 Introduction

This code reimplements the footnote interfaces for \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^AT_EX 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^AT_EX'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 mark¹ 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:

¹Like 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`, `sclttr2`, 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).

fntext/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

fntext/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**.

fntext

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.

fntext/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.

fntext/begin & fntext/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.

fntext/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 cname changes)*

Footnotes consist of a *footnotemark* (short: mark) that is typically placed in the text as a superscript number like this¹, 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 mark¹⁻³ 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: Their name and/or function will probably change as they currently mix tagging with the link support.

TODO: review this sockets

tagsupport/fnmark (1 argument)

The socket is used in `\@footnotemark/\fnote_footnotemark:` and takes `\@makefnmark` as argument. It prints the mark in the text and surrounds it with a tagging structure and a link. As such it is not solely for tagging and so should not be used with `\UseTaggingSocket` as this would swallow the argument and lose the link support.

TODO: *describe and decide on names*

tagsupport/fntext/begin (no argument)

This socket is used before the main processing socket (so before the `\insert` command). It opens the FEnote structure. As it sets also the `tl-var` for the current structure and this is used in destinations it should not use as tagging socket.

tagsupport/fntext/end (no argument)

This socket is used after the main processing socket (so after the `\insert` command). It closes the FEnote structure.

tagsupport/fntext/mark (1 argument)

This socket is used around the mark in the footnote text. It adds tagging support but also link support, so like the other tagsupport sockets it should be always active.

tagsupport/fntext/text (1 argument)

This socket handles mc-chunks around the text of the footnote. As it takes an argument (the text) it should not be used as tagging socket either.

The *footnotemark* should create a `/Lb1` structure² that should contain a `/Ref` entry pointing to the structure of the *footnotetext*.

The *footnotetext* should create a `/FENote`³ 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`⁴ 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 wants to avoid faulty nesting.
- It must detect which marks are related to which notes so that it can setup the `/Ref` cross-references.

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

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

⁴We tag it as `/footnotelabel`.

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.⁵

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 one or more match it stores the structure numbers and removes the entries from the property (so in a normal document the property will never contain more than a few entries).

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`

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

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.

<code>\fnote_mark_gput:nn</code>	<code>\fnote_mark_gput:nn{<mark>}{<class name>}</code>
<code>\fnote_mark_gput:(no oo)</code>	

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

<code>\fnote_mark_gpop_all:nnN</code>	<code>\fnote_mark_gpop_all:nnN{<mark>}{<class name>}{<sequence>}</code>
---------------------------------------	---

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.

⁵Currently `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.

3.3.1 `\footref`

`\footref` uses 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 `tagsupport/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 `tagsupport/fntext/begin` and `tagsupport/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^AT_EX 2_ε name `\@makefntext`). In the new implementation this contains two further kernel sockets for tagging: `tagsupport/fntext/mark` and `tagsupport/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 `\@makefntext`.

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 %       \@makefnmark{%
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.)

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.)

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.)

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 {2}
```

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^AT_EX 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.)

\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.)

5.8 Hooks

fnmark/before (hook) Hooks in the footnotemark command.

```

fnmark/after (hook)
    fnmark (hook) 107 \NewMirroredHookPair{fnmark/before}{fnmark/after}
fnmark/begin (hook) 108 \NewHook{fnmark}
    fnmark/end (hook) 109 \NewHook{fnmark/begin}
    110 \NewHook{fnmark/end}
```

fntext/before (hook) Hooks in the footnotetext command:

```

fntext/after (hook)
    fntext (hook) 111 \NewMirroredHookPair{fntext/before}{fntext/after}
fntext/begin (hook) 112 \NewHook{fntext}
    fntext/end (hook) 113 \NewHook{fntext/para}
    fntext/para (hook) 114 \NewHook{fntext/begin}
    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.)

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 {tagsupport/fnmark}
144         \socket_log:n {tagsupport/fntext/begin}
145         \socket_log:n {tagsupport/fntext/end}
146         \socket_log:n {tagsupport/fntext/mark}
147         \socket_log:n {tagsupport/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}
153         \hook_log:n {fntext/end}
154         \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_footnotemark_finish:` below does nothing and the action is carried out later.

```

180   \__fnote_footnotemark_finish:
181 }

```

(End of definition for \fnote_footnotemark:.)

`__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:.`)

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

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

`\@footnotemark` Here we provide the traditional L^AT_EX 2_ε name in case it is directly used in some legacy class.

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

(End of definition for `\@footnotemark.`)

5.11 The new `\@footnotetext` command

`\fnote_footnotetext:n` We temporarily test for the tagging socket until it is in the next release:

```
200 \socket_if_exist:nF {tagssupport/para/restore}
201 {
202   \NewSocket{tagssupport/para/restore}{0}
203   \NewSocketPlug{tagssupport/para/restore}{default}
204   {
205     \tl_set:Nn \l__tag_para_main_tag_tl {text-unit}
206     \tl_set_eq:NN \l__tag_para_tag_tl \l__tag_para_tag_default_tl
207     \bool_set_false:N \l__tag_para_flattened_bool
208   }
209   \AssignSocketPlug{tagssupport/para/restore}{default}
210 }

211 \cs_new_protected:Npn \fnote_footnotetext:n #1 {
212   \__fnote_debug_footnotetext:
213   %-----
214   % ./linguex/linguex.sty
215   \hook_use:n {fntext/before}
216   %-----
```

Execute a kernel socket for tagging.

```
217   \socket_use:n {tagssupport/fntext/begin}

218   \socket_use:nn {fntext/process}
219   {
220     %-----
221     % resetting baselinestretch ... (could be done further down)
222     % ./uafthesis/uafthesis.cls
223     % ./setspace/setspace.sty
224     % ./footmisc/footmisc.sty (normal)
225     \hook_use:n {fntext}
226     %-----
227     \reset@font
228     \footnotesize
229     %-----
230     % some classes use a different font size, e.g.,
```

```

231 % ./nrc/nrc1.cls ./nrc/nrc2.cls
232 % but those could be done in fntext/para instead
233 %-----

```

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

```

234 \interlinepenalty\interfootnotelinepenalty
235 \splittopskip\footnotesep
236 \splitmaxdepth \dp\strutbox
237 \floatingpenalty \@MM
238 \hsize\columnwidth
239 \@parboxrestore
240 \UseTaggingSocket{para/restore}
241 \parindent 1em % typical default used in \@makefntext moved up here
242 \def\@currentcounter{footnote}
243 \protected@edef \@currentlabel { \p@footnote \@thefnmark }
244 %-----
245 % for altering para parameters ...
246 % code for resphilosophica came earlier but it could go here.
247 % Has the advantage that one can also overwrite \cs{@currentcounter}
248 % and \cs{@currentlabel} is that is necessary.
249 %
250 % ./resphilosophica/resphilosophica.cls
251 \hook_use:n {fntext/para}
252 %-----
253 \color@begingroup
254 %-----
255 % fnpara wants to replace \@makefntext{...} and para and side
256 % option of footmisc etc too ...
257 % so we make this a socket, because only one action can be active:
258 %-----
259 \socket_use:nn {fntext/make}
260 {
261 %-----
262 % ./resphilosophica/resphilosophica.cls
263 %-----
264 \socket_use:n {fntext/begin}%
265 %-----
266 % bibarts
267 % fnbreak.sty
268 \hook_use:n {fntext/begin}
269 %-----
270 \ignorespaces
271 #1
272 %-----
273 % bibarts
274 % fnbreak.sty
275 \hook_use:n {fntext/end}
276 %-----

```

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

hook above.

```

277         \socket_use:n {fntext/end}
278     }
279     \par
280     \color@endgroup
281 }
282 %-----

```

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

```

283     \socket_use:n{tagssupport/fntext/end}
284 %-----
285 % ./linguex/linguex.sty
286     \hook_use:n {fntext/after}
287 %-----
288 }

```

(End of definition for `\fnote_footnotetext:n`.)

fntext/process (*socket*)

```

289 \NewSocket      {fntext/process}{1}
290 \NewSocketPlug{fntext/process}{default}{ \insert\footins {#1} }
291 \NewSocketPlug{fntext/process}{side}   { \marginpar {#1} }

292 \AssignSocketPlug{fntext/process}{default}

```

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

```

293 \NewSocket      {fntext/make}{1}
294 \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).

```

295 \NewSocketPlug{fntext/make}{para}
296 {
297     \setbox\FN@tempboxa\hbox{\@makefntext{#1}}%
298     \dp\FN@tempboxa\z@
299     \ht\FN@tempboxa
300     \dimexpr\wd\FN@tempboxa *%
301         \footnotelineskip /\columnwidth\relax
302     \box\FN@tempboxa
303 }

304 \AssignSocketPlug{fntext/make}{default}

```

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

```

305 \NewSocket      {fntext/begin}{0}
306 \NewSocketPlug{fntext/begin}{default}{ \rule\z@\footnotesep }

```

```
307 \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.

```
308 \NewSocket      {fntext/end}{0}
309 \NewSocketPlug{fntext/end}{default}{ \@finalstrut\strutbox }
```

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

```
310 \NewSocketPlug{fntext/end}{para}
311   {%
312       \strut
313       \penalty-10\relax
314       \hskip\footglue
315   }

316 \AssignSocketPlug{fntext/end}{default}
```

`tagssupport/fntext/begin (socket)` Kernel sockets for tagging.
`tagssupport/fntext/end (socket)`

```
317 \NewSocket{tagssupport/fntext/begin}{0}
318 \NewSocket{tagssupport/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).

```
319 \AddToHook{begindocument}
320 {
321     \cs_set_eq:NN \@footnotetext \fnote_footnotetext:n
322 }
```

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.

```
323 \newdimen\footnotemargin
324 \footnotemargin\maxdimen          % no value given
325
326 \AtBeginDocument
327 {
328     \ifdim \footnotemargin=\maxdimen
329         \setlength\footnotemargin{1.8em}
330     \fi
331 }
```

`\fnote_makefntext:n`

```
332 \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.

```

333 \noindent

334 \socket_use:nn {tagssupport/fntext/mark} { \socket_use:n {fntext/mark} }
335 \socket_use:nn {tagssupport/fntext/text} { \socket_use:nn {fntext/text}{#1} }
336 }

```

(End of definition for `\fnote_makefntext:n`.)

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

```

337 \NewSocket {fntext/mark}{0}

```

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

```

338 \NewSocketPlug{fntext/mark}{default}{
339   \ifdim\footnotemargin>\z@
340     \hb@xt@ \footnotemargin{\hss\@makefnmark}
341   \else
342     \ifdim\footnotemargin=\z@
343       \llap{\@makefnmark}
344     \else
345       \ifdim\footnotemargin=-\maxdimen
346         \@makefnmark
347       \else
348         \llap{\hb@xt@ -\footnotemargin{\@makefnmark\hss}}
349       \fi
350     \fi
351   \fi
352 }

353 \AssignSocketPlug{fntext/mark}{default}

```

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

```

354 \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.

```

355 \NewSocket{tagssupport/fntext/mark}{1}
356 \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`

```

357 \newcommand\old@std@class@makefntext[1]{%
358   \parindent 1em%
359   \noindent
360   \hb@xt@1.8em{\hss\@makefnmark}\#1}

```

(End of definition for \old@std@class@makefntext.)

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 `to 1em` or similar. It then replaces that with

```
{AAA \UseSocket{tagsupport/fntext/mark}{\hbox BBB { CCC }} DDD
\UseSocket{tagsupport/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).

```

361 \tl_new:N \l__fnote_patch_tl
362 \cs_new_eq:NN \__fnote_tmp:w \ERROR
363
364 \cs_new_protected:Npn \__fnote_patch:
365 {
366   \tl_set:Nn \l__fnote_patch_tl { \@makefntext { \UseSocket{tagsupport/fntext/text}{\#1} } }
367   \tl_if_in:NnTF \l__fnote_patch_tl { \hbox }
368   { \cs_set_eq:NN \__fnote_tmp:w \__fnote_patch_hbox:w }
369   {
370     \tl_if_in:NnTF \l__fnote_patch_tl { \hb@xt@ }
371     { \cs_set_eq:NN \__fnote_tmp:w \__fnote_patch_hb@xt:w }
372     {

```

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

```

373   \tl_if_in:NnTF \l__fnote_patch_tl { \makebox }
374   { \cs_set_eq:NN \__fnote_tmp:w \__fnote_patch_makebox:w }
375   {
376     \tl_if_in:NnTF \l__fnote_patch_tl { \@makefnmark }
377     { \cs_set_eq:NN \__fnote_tmp:w \__fnote_patch_@makefnmark:w }
378     { \ERROR
379       \cs_set_eq:NN \__fnote_tmp:w \exp_stop_f: }

```

```

380     }
381   }
382 }
383 \tl_set:Nf \l__fnote_patch_tl
384 { \exp_after:wN \__fnote_tmp:w \l__fnote_patch_tl }
385 \cs_set:Npn \__fnote_tmp:w { \long \def \@makefntext ###1 }
386 \exp_after:wN \__fnote_tmp:w \exp_after:wN { \l__fnote_patch_tl }
387 }

```

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

```

AAA \@makefntext@processX { \hbox BBB }

388 \cs_new:Npn \__fnote_patch_hbox:w #1 \hbox #2 #
389 { \exp_stop_f: #1 \@makefntext@processX { \hbox #2 } }

```

Same for the other cases.

```

390 \cs_new:Npn \__fnote_patch_hb@xt@:w #1 \hb@xt@ #2 #
391 { \exp_stop_f: #1 \@makefntext@processX { \hb@xt@ #2 } }

392 \cs_new:Npn \__fnote_patch_makebox:w #1 \makebox #2 #
393 { \exp_stop_f: #1 \@makefntext@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.

```

394 \cs_new:Npn \__fnote_patch_@makefnmark:w #1 \@makefnmark
395 { \exp_stop_f: #1 \@makefntext@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:

```

396 \cs_new:Npn \@makefntext@processX #1#2{\UseSocket{tagssupport/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.

```

397 \AddToHook{begindocument}
398 {
399   \cs_if_eq:NNTF \@makefntext \old@std@class@makefntext
400   {
401     \cs_set_eq:NN \@makefntext \fnote_makefntext:n
402   }
403   {

```

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

```

404     \cs_if_eq:NNTF \@makefntext \footmisc@hang@makefntext
405     { \__fnote_patch: }
406   }
407 }

```



```

408
409
410 % possibly add the following to check for multiple \hbox in
411 % the definition:
412 %
413 % \seq_set_split:NnV \l__fnote_patch_seq { \hbox } \l__fnote_patch_tl
414 % \int_compare:nT { \seq_count:N \l__fnote_patch_seq } > 2 \ERROR
415 %

```

5.13 Document-level commands

\footnotetext

```

416 \DeclareDocumentCommand\footnotetext {o+m}
417 {
418   \fnote_set_fnmark:nn {#1} \@mpfn
419   \@footnotetext {#2}
420 }

```

(End of definition for \footnotetext.)

\footnote

```

421 \DeclareDocumentCommand\footnote {o+m}
422 {
423   \fnote_step_fnmark:nn {#1} \@mpfn
424   \cs_set_eq:NN \__fnote_footnotemark_finish: \prg_do_nothing:
425   \@footnotemark
426   \cs_set_eq:NN \__fnote_footnotemark_finish: \__fnote_footnotemark_default_finish:
427   \@footnotetext {#2}
428   \__fnote_footnotemark_finish:
429 }

```

(End of definition for \footnote.)

\footnotemark

```

430 \DeclareDocumentCommand\footnotemark {o}
431 {
432   \fnote_step_fnmark:nn {#1} { footnote }
433   \@footnotemark
434 }

```

(End of definition for \footnotemark.)

\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.

```

435 \DeclareDocumentCommand\footref {m}
436 {
437   \begingroup
438   \unrestored@protected@xdef\@thefnmark{\ref*{#1}}%

```

```

439 \endgroup
440 \bool_set_false:N \l__fnote_autodetect_bool
441 \tl_set:Nn \l__fnote_currentlabel_tl {#1}
442 \@footnotemark
443 \bool_set_true:N \l__fnote_autodetect_bool
444 }

```

(End of definition for `\footref`.)

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.

```

445 \NewSocketPlug{fntext/process}{mp}
446 {
447   \global\setbox\@mpfootins\vbox{%
448     \unvbox\@mpfootins
449     #1
450   }
451 }

```

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.

```

452 \AddToHook{class/memoir/before}
453 { \let\new@std@class@makecol\@makecol }
454 \AddToHook{class/memoir/after}
455 {
456   \cs_set_eq:NN \@footnotemark \fnote_footnotemark:
457   \cs_set_eq:NN \@makefntext\old@std@class@makefntext
458   \cs_set_eq:NN \@makecol\new@std@class@makecol
459 }

```

5.15.2 setspace

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

```

460 \AddToHook{package/setspace/after}
461 {\let \@footnotetext \fnote_footnotetext:n
462   \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.

```

463 \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.

```

464 \AtBeginDocument
465 {
466   \providecommand\hyper@linkstart{\@gobbletwo}
467   \providecommand\hyper@linkend{\@empty}
468 }

```

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).

```

469 \AtBeginDocument
470 {
471   \ifpackageloaded{hyperref}
472   {
473     \legacy_if:nF{Hy@hyperfootnotes}{\bool_set_false:N \l_fnote_link_bool}
474   }
475   {
476     \bool_set_false:N \l_fnote_link_bool
477   }
478 }

```

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`

```

479 \property_new:nmn {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
```

```

480 \cs_new_protected:Npn \__fnote_label_hook:e #1 {}
481 \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.

```

482 \AddToHook{fntext/begin}
483 {
484   \cs_set_protected:Npn \__fnote_label_hook:e #1
485   {
486     \property_record:ee {\__fnote/#1} {fnote/struct}
487   }
488 }

```

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!

```

489 \cs_new_protected:Npn \fnote_class_new:nn #1 #2 % #1 name, #2 options
490 {
491   \prop_new:c { g__fnote_currentmarks_ #1 _prop }
492 }
493
494 \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).

```

495 \cs_new_protected:Npn \fnote_mark_gput:nn #1 #2 % #1 the representation of the mark, #2 type
496 {
497   \prop_gput:cen { g__fnote_currentmarks_ #2 _prop }
498     { \tag_get:n{struct_num} }
499     { #1 }
500 }
501
502 \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.

```

503 \cs_new_protected:Npn \fnote_mark_gpop_all:nnN #1 #2 #3
504 {
505   \seq_clear:N #3
506   \prop_set_eq:Nc \l__fnote_tmpa_prop { g__fnote_currentmarks_ #2 _prop }
507   \prop_map_inline:Nn \l__fnote_tmpa_prop
508     {
509       \tl_if_eq:nnT {#1} { ##2 }
510       {

```

store the key (the structure number) in the seq

```

511       \seq_put_right:Nn #3 { ##1 }

```

remove entry as used from the global prop

```

512         \prop_gremove:cn { g__fnote_currentmarks_ #2 _prop } {##1}
513     }
514 }
515 }
516 \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 `\@makefnmark` as its argument. At this time `\@thefnmark` is already set.

```

517 \NewSocketPlug{tagsupport/fnmark}{FEMark}
518 {

```

End an open mc and start the structure.

```

519     \tag_mc_end_push:
520     \tag_struct_begin:n { tag=footnotemark }

```

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

```

521     \bool_if:NTF \l__fnote_autodetect_bool
522     {

```

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.

```

523         \fnote_mark_gput:oo {\@thefnmark}{\l__fnote_type_tl}
524         \tl_set:Nc \l__fnote_linktarget_tl {footnote*.\tag_get:n{struct_num}}
525     }

```

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 /Ref of the FEnote. Both must be delayed as we don't know if the objects of the FEnote and the mark have already been created.

```

526     {
527         \hook_gput_code:nne {tagpdf/finish/before} {tagpdf/footnote}
528         {
529             \exp_not:N\fnote_gput_refs:ee
530             { \tag_get:n{struct_num} }
531             { \property_ref:ee{ __fnote/\l__fnote_currentlabel_tl } {fnote/struct} }
532         }

```

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

```

533         \tl_set:Nc \l__fnote_linktarget_tl {footnote*.\property_ref:ee {__fnote/\l__fnote_cu
534     }

```

And now the actual content

```

535 \tag_mc_begin:n{tag=Lbl}
536 %
537 \bool_if:NTF \l_fnote_link_bool
538 {
539     \exp_args:No
540     \hyper@linkstart
541     { \l_fnote_link_type_tl }
542     { \l__fnote_linktarget_tl }
543     #1
544     \hyper@linkend
545 }
546 { #1 }
547 \tag_mc_end:
548 \tag_struct_end:
549 \tag_mc_begin_pop:n{ }
550 }

```

At last assign the plug:

```

551 \AssignSocketPlug{tagssupport/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
552 \cs_new_protected:Npn \__fnote_gput_ref:nn #1 #2 % #1 the structure number receiving the ref
553 {
554     \tag_struct_gput:nnn {#1}{ref_num}{#2}
555 }
556 \cs_new_protected:Npn \fnote_gput_refs:nn #1 #2 % pair of numbers
557 {
558     \__fnote_gput_ref:nn {#1}{#2}
559     \__fnote_gput_ref:nn {#2}{#1}
560 }
561 \cs_generate_variant:Nn \fnote_gput_refs:nn {ee}

```

(End of definition for __fnote_gput_ref:nn and \fnote_gput_refs:nn.)

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

FENote (*plug*)

```

562 \NewSocketPlug{tagssupport/fntext/begin}{FENote}
563 {
564     \tag_mc_end_push:

```

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

```

565 \tag_check_child:nnTF {FENote}{pdf2}
566 {
567     \tag_struct_begin:n { tag=footnote }

```

```

568     }
569     {
570         \tag_struct_begin:n
571         {
572             tag=footnote,

```

We add 0 for now to ensure to get a number even if the sec code is not loaded or if the seq is empty (which it shouldn't unless there is an coding error)

```

573         parent=\int_max:nn{2}{\tag_get:n{current_Sect}+0}
574     }
575 }

```

Store the current structure number for labels.

```

576     \tl_set:Nx \l__fnote_currentstruct_tl { \tag_get:n{struct_num} }

```

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.

```

577     \bool_if:NTF \l__fnote_autodetect_bool
578     {

```

find open marks with identical \@thefnmark

```

579         \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:

```

580         \seq_map_inline:Nn \l__fnote_currentrefs_seq
581         {
582             \fnote_gput_refs:ee {##1}{ \l__fnote_currentstruct_tl }
583         }
584     }

```

If no auto detection is done

```

585         {%no auto
586
587         }
588     }

```

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.

```

589 \NewSocketPlug{taggsupport/fntext/end}{FENote}
590 {
591     \tag_struct_end:
592     \tag_mc_begin_pop:n{
593 }

```

At last assign the plugs:

```
594 \AssignSocketPlug{tagsupport/fntext/begin}{FENote}
595 \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.

```
596 \NewSocketPlug{tagsupport/fntext/mark}{FENoteLbl}
597 {
598   \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`).

```
599   %\seq_show:N\l__fnote_currentrefs_seq
600   \seq_map_inline:Nn\l__fnote_currentrefs_seq {\MakeLinkTarget*{footnote*.\##1}}
601   \MakeLinkTarget*{footnote*.\l__fnote_currentstruct_tl}
```

Now we add the tagging commands. We move the structure of the label to the begin of the footnote structure.

```
602   \tag_struct_begin:n { tag=footnotelabel,parent=\l__fnote_currentstruct_tl,firstkid }
603   \tag_mc_begin:n { tag=Lbl }
604   #1
605   \tag_mc_end:
606   \tag_struct_end:
607   \tag_mc_begin_pop:n{
608 }

609 \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?

```
610 \NewSocketPlug{tagsupport/fntext/text}{FENotetext}
611 {
612   \tag_mc_end_push:
613   \tag_mc_begin:n{
614     #1
615     \tag_mc_end:
616     \tag_mc_begin_pop:n{
617 }

618 \AssignSocketPlug{tagsupport/fntext/text}{FENotetext}
```

```

619 \ExplSyntaxOff
620 </kernel>

621 <@@=>

```

6 Reimplementing the footmisc package

```

622 <{*footmisc}
623 %%
624 %% Copyright (c) 1995–2011 Robin Fairbairns
625 %% Copyright (c) 2018–2023 Robin Fairbairns, Frank Mittelbach
626 %%
627 %% This file is part of the `latex-lab Bundle'.
628 %% -----
629 %%
630 %% It may be distributed and/or modified under the
631 %% conditions of the LaTeX Project Public License, either version 1.3c
632 %% of this license or (at your option) any later version.
633 %% The latest version of this license is in
634 %%   https://www.latex-project.org/lppl.txt
635 %% and version 1.3c or later is part of all distributions of LaTeX
636 %% version 2008 or later.
637 %%
638 %% This work has the LPPL maintenance status 'maintained'.
639 %%
640 \NeedsTeXFormat{LaTeX2e}
641 \providecommand\DeclareRelease[3]{}
642 \providecommand\DeclareCurrentRelease[2]{}
643
644 \DeclareRelease{v5}{2011-06-06}{footmisc-2011-06-06.sty}
645 \DeclareCurrentRelease{}{2022-02-14}
646 \ProvidesPackage{latex-lab-footmisc}%
647   [2025/02/20 v6.0e
648   a miscellany of footnote facilities -- latex-lab version%
649   ]
650
651 \NeedsTeXFormat{LaTeX2e}[2020/10/01]
652 \newtoks\FN@temptoken
653 \providecommand\protected@writeaux{%
654   \protected@write\@auxout
655 }
656 \def\l@advance@macro{\@@d@vance@macro\edef}
657 \def\@@d@vance@macro#1#2#3{\expandafter\@tempcnta#2\relax
658   \advance\@tempcnta#3\relax
659   #1#2{\the\@tempcnta}%
660 }
661 \let\@advance@macro\l@advance@macro
662 \DeclareOption{symbol}{\renewcommand\thefootnote{\fnsymbol{footnote}}}
663 \newif\ifFN@robust \FN@robustfalse
664 \DeclareOption{symbol*}{%
665   \renewcommand\thefootnote{\@fnsymbol\c@footnote}%
666   \FN@robusttrue

```

```

667 \AtEndOfPackage{\setfnsymbol{lamport*--robust}}%
668 }
669 \newif\ifFN@para \FN@parafalse
670 \DeclareOption{para}{%

```

Options are executed in the order of declaration, thus no point in checking for side option as footmisc did in the past

```

671 % \PackageError{footmisc}{Option "\CurrentOption" incompatible with
672 % option "side"}%
673 % {I shall ignore "\CurrentOption"}%
674 \FN@paratrue
675 \setlength\footnotemargin{-\maxdimen} % default when para is used
676 }

677 \DeclareOption{side}{\ifFN@para
678 \PackageError{footmisc}{Option "\CurrentOption" incompatible with
679 option "para"}%
680 {I shall ignore "\CurrentOption"}%
681 \else
682 \AddToHook{fntext/para}{%
683 \hspace\marginparwidth % correct the default \hspace
684 \footnotesep\z@ % don't add a default separation
685 }
686 \AssignSocketPlug{fntext/process}{side}
687 % \AssignSocketPlug{fntext/make}{default}
688 \AssignSocketPlug{fntext/begin}{noop}
689 \AssignSocketPlug{fntext/end}{noop}
690 \fi
691 }
692 \let\footnotelayout\@empty
693 \DeclareOption{ragged}{%
694 \@ifundefined{RaggedRight}%
695 {\renewcommand\footnotelayout{\linepenalty50 \raggedright}}%
696 {\renewcommand\footnotelayout{\linepenalty50 \RaggedRight}}%
697 }
698 \newif\ifFN@perpage
699 \FN@perpagefalse
700 \DeclareOption{perpage}{%
701 \FN@perpagetrue
702 }
703 \newif\ifFN@fixskip \FN@fixskipfalse
704
705 \let\FN@bottomcases\thr@@
706 \newif\ifFN@abovefloats \FN@abovefloatstrue
707 \DeclareOption{bottom}{%
708 \let\FN@bottomcases\@ne
709 \FN@abovefloatsfalse
710 \FN@fixskiptrue
711 }
712 \DeclareOption{bottomfloats}{%
713 \let\FN@bottomcases\tw@
714 \FN@abovefloatstrue \FN@fixskiptrue
715 }
716 \DeclareOption{abovefloats}{\FN@abovefloatstrue \FN@fixskiptrue}
717 \DeclareOption{belowfloats}{\FN@abovefloatsfalse \FN@fixskiptrue}

```

```

718 \DeclareOption{marginal}{%
719   \footnotemargin-0.8em\relax
720 }
721 \DeclareOption{flushmargin}{%
722   \footnotemargin0pt\relax
723 }
724 \newif\ifFN@hangfoot \FN@hangfootfalse
725 \DeclareOption{hang}{%
726   \FN@hangfoottrue
727 }
728 \newcommand*\hangfootparskip{0.5\baselineskip}
729 \newcommand*\hangfootparindent{0em}%
730 \DeclareOption{norule}{%
731   \renewcommand\footnoterule{%
732     \advance\skip\footins 4\p@\@plus2\p@\relax
733   }
734 \DeclareOption{splitrule}{%
735   \gdef\split@prev{0}
736   \let\pagefootnoterule\footnoterule
737   \let\mpfootnoterule\footnoterule
738   \def\splitfootnoterule{\kern-3\p@ \hrule \kern2.6\p@}
739   \def\footnoterule{\relax
740     \ifx \@listdepth\@mplistdepth
741       \mpfootnoterule
742     \else
743       \ifnum\split@prev=\z@
744         \pagefootnoterule
745       \else
746         \splitfootnoterule
747       \fi
748       \xdef\split@prev{\the\insertpenalties}%
749     \fi
750   }%
751 }
752 \newif\ifFN@stablefootnote \FN@stablefootnotefalse
753 \DeclareOption{stable}{\FN@stablefootnotetrue}
754 \newif\ifFN@multiplefootnote \FN@multiplefootnotefalse
755 \DeclareOption{multiple}{\FN@multiplefootnotetrue}
756 \ProcessOptions

```

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

```

757 \ifFN@para
758   \NewSocketPlug{build/column/footnotes}{para}{%
759     \global\setbox\footins\vbox{\FN@makefootnoteparagraph}%
760   }
761   \AssignSocketPlug{build/column/footnotes}{para}
762 \fi
763 \ifFN@fixskip
764   \def\@outputbox@removebskip{%
765     \ifx\@textbottom\relax \else
766       \@outputbox@append{%
767         \@tempskipa\lastskip
768         \ifnum \gluestretchorder\@tempskipa>\z@

```

```

769         \vskip-\@tempskipa
770         \xdef\@outputbox@reinsertbskip
771             {\noexpand\@outputbox@append{\vskip\the\@tempskipa}}}%
772     \else
773         \global\let\@outputbox@reinsertbskip\relax
774     \fi
775 }%
776 \fi
777 }
778 \let\@outputbox@reinsertbskip\relax
779 \else
780     \let\@outputbox@removebskip \relax
781     \let\@outputbox@reinsertbskip\relax
782 \fi

783 \ifcase \FN@bottomcases\relax
784     \ERROR
785 \or %1 bottom option
786     \ifFN@abovefloats
787         \AssignSocketPlug {build/column/outputbox}{space-footnotes-floats}
788     \else
789         \AssignSocketPlug {build/column/outputbox}{floats-space-footnotes}
790     \fi
791 \or %2 bottomfloats
792     \ifFN@abovefloats
793         \AssignSocketPlug {build/column/outputbox}{footnotes-space-floats}
794     \else
795         \AssignSocketPlug {build/column/outputbox}{space-floats-footnotes}
796     \fi
797 \or %3 neither bottom nor bottomfloats
798     \ifFN@abovefloats
799         \AssignSocketPlug {build/column/outputbox}{footnotes-floats}
800     \else
801         \AssignSocketPlug {build/column/outputbox}{floats-footnotes}
802     \fi
803 \else
804     \ERROR
805 \fi

806
807 % next can be dropped when cleaned up
808 \newif\ifFN@setspace
809 \@ifpackageloaded{setspace}%
810 {%
811     \FN@setspacetrue
812     \@ifclassloaded{memoir}%
813     {%
814         \AddToHook{fntext}{\let\baselinestretch\m@m@singlespace}%
815         \let\FN@baselinestretch\m@m@singlespace
816     }%
817     {%
818         \AddToHook{fntext}{\let\baselinestretch\setspace@singlespace}%
819         \let\FN@baselinestretch\setspace@singlespace
820     }%
821 }%
822 {%

```

```

823 \FN@setspacefalse
824 }
825
826
827
828 \ifFN@para
829
830 % \AssignSocketPlug{fntext/process}{default}
831 \AssignSocketPlug{fntext/make}{para}
832 \AssignSocketPlug{fntext/begin}{noop}
833 \AssignSocketPlug{fntext/end}{para}
834
835 \fi
836
837
838
839 \ifFN@para
840 \let\FN@tempboxa\@tempboxa
841 \newbox\FN@tempboxb
842 \newbox\FN@tempboxc
843 \newskip\footglue \footglue=1em plus.3em minus.3em
844
845 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
846 \newdimen\footnotebaselineskip
847
848 % establish late:
849
850 \AddToHook{begindocument/before} {%
851 {%
852 \footnotesize
853 \global\footnotebaselineskip=\normalbaselineskip
854 }%
855 }

```

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.

```

856 \long\def\FN@makefootnoteparagraph{%
857 \FN@setfootnoteparawidth
858 \@parboxrestore
859 \baselineskip=\footnotebaselineskip
860 \unvbox\footins \FN@removehboxes
861 \RawParEnd
862 }
863 \def\FN@removehboxes{\setbox\FN@tempboxa\lastbox
864 \ifhbox\FN@tempboxa{\FN@removehboxes}%
865 \unhbox\FN@tempboxa
866 \else
867 \RawNoindent
868 \rule{z@}{footnotesep}
869 \fi
870 }
871 \fi
872
873

```

```

874 \ifpackageloaded{multicol}
875   {\def\FN@setfootnoteparawidth
876     {\hsize\ifnum\doublecol@number>\@ne
877       \textwidth
878       \else \columnwidth \fi}}
879   {\def\FN@setfootnoteparawidth{\hsize\columnwidth}}
880
881 \ifFN@perpage
882   \RequirePackage{perpage}
883   \MakePerPage{footnote}

```

Fix a bug in perpage ...

```

884 \def\@stpelt#1{\global\csname c@#1\endcsname \m@ne
885   \stepcounter{#1}%
886   \pp@fix@MakePerPage{#1}%
887 }
888 \def\pp@fix@MakePerPage#1{%
889   \ifnum \value{#1}>\z@
890     \addtocounter{#1}\m@ne\fi
891 }

```

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:

```

892 \def\pp@c1@end@iii\stepcounter#1\pp@fix@MakePerPage#2{
893 \fi
894
895
896 \ifFN@para
897
898 % This can use the default interface, except that a negative value for
899 % \footnotemargin makes little sense, so we test for this and warn if
900 % necessary. But -\maxdimen is ok again, so would need to be a little bit more elaborate.
901 %
902
903 %\AddToHook{fntext/para}{
904 % \ifdim \footnotemargin >\z@ \else
905 %   \PackageWarningNoline{footmisc}{Option 'para' needs positive \noexpand\footnotemargin}%
906 %   \footnotemargin 1.8em\relax
907 % \fi
908 %}
909
910
911 \AddToHook{fntext/begin}{\nobreak \hspace{.2em}}
912
913 \else
914
915 \ifFN@hangfoot
916   \long\def\footmisc@hang@makefntext#1{%
917     \bgroup

```

```

918 \SuspendTagging{footmisc}%
919 \setbox\@tempboxa\hbox{%
920 \ifdim\footnotemargin>\z@
921 \hb@xt@\footnotemargin{\@makefnmark\hss}%
922 \else
923 \@makefnmark
924 \fi
925 }%
926 \leftmargin\wd\@tempboxa
927 \rightmargin\z@
928 \linewidth \columnwidth
929 \advance \linewidth -\leftmargin
930 \parshape \@ne \leftmargin \linewidth
931 \footnotesize
932 \@setpar{\@par}}%
933 \ResumeTagging{footmisc}%
934 \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.

```

935 \UseSocket{tagsupport/fntext/mark}%
936 {\llap{%
937 \ifdim\footnotemargin>\z@
938 \hb@xt@\footnotemargin{\@makefnmark\hss}%
939 \else
940 \@makefnmark
941 \fi
942 }}%
943 \parskip\hangfootparskip\relax
944 \parindent\hangfootparindent\relax
945 \footnotelayout#1%
946 \par
947 \egroup
948 }

```

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

```

949 \let \@makefntext \footmisc@hang@makefntext
950
951 \else
952
953 % This is now using the default interface:
954 %
955 % \long\def\@makefntext#1{%
956 % \parindent1em
957 % \noindent
958 % \ifdim\footnotemargin>\z@
959 % \hb@xt@ \footnotemargin{\hss\@makefnmark}%
960 % \else
961 % \ifdim\footnotemargin=\z@
962 % \llap{\@makefnmark}%
963 % \else
964 % \llap{\hb@xt@ -\footnotemargin{\@makefnmark\hss}}%

```

```

965 %          \fi
966 %          \fi
967 %          \footnotelayout#1%
968 %      }
969
970 \fi
971 \fi
972
973
974
975
976 \ifFN@multiplefootnote
977 \providecommand*{\multiplefootnotemarker}{3sp}

We tag the separator as artifact
TODO: why is this done with \providecommand?

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

```



```

1016 }
1017 \def\FN@sf@gobble@twobrace[#1]#2{}
1018 \let\FN@sf@@footnotemark\footnotemark
1019 \def\footnotemark{\ifx\protect\@typeset@protect
1020   \expandafter\FN@sf@@footnotemark
1021   \else
1022     \expandafter\FN@sf@gobble@optonly
1023   \fi
1024 }
1025 \edef\FN@sf@gobble@optonly{\noexpand\protect
1026   \expandafter\noexpand\csname FN@sf@gobble@optonly \endcsname}
1027 \expandafter\def\csname FN@sf@gobble@optonly \endcsname{%
1028   \@ifnextchar[%]
1029     \FN@sf@gobble@bracket
1030     {}%
1031 }
1032 \def\FN@sf@gobble@bracket[#1]{}
1033 \fi
1034 \newcommand\setfnsymbol[1]{%
1035   \@bsphack
1036   \@ifundefined{FN@fnsymbol@#1}%
1037   {%
1038     \PackageError{footmisc}{Symbol style "#1" not known}%
1039     \@eha
1040   }{%
1041     \expandafter\let\expandafter\@fnsymbol\csname
1042       FN@fnsymbol@#1\endcsname
1043   }%
1044   \@esphack
1045 }
1046 \let\FN@fnsymbol@lamport\@fnsymbol
1047 \newif\if@tempswb
1048 \DeclareDocumentCommand\DefineFNSymbols {sm0{text}m}{%
1049   \expandafter\ifx\csname FN@fnsymbol@#2\endcsname\relax
1050     \PackageInfo{footmisc}{Declaring symbol style #2}%
1051   \else
1052     \PackageWarning{footmisc}{Redeclaring symbol style #2}%
1053   \fi
1054   \toks@{}%
1055   \def\@tempb{\end}%
1056   \FN@build@symboldef#4\end
1057   \def\@tempc{math}%
1058   \def\@tempd{#3}%
1059   \expandafter\xdef\csname FN@fnsymbol@#2\endcsname##1{%
1060     \ifx\@tempc\@tempd
1061       \noexpand\ensuremath
1062     \else
1063       \noexpand\nfss@text
1064     \fi
1065     {%
1066       \noexpand\ifcase##1%
1067       \the\toks@
1068       \noexpand\else
1069       \IfBooleanTF#1{\noexpand\@ctrerr}%

```

```

1070         {\noexpand\FN@orange##1}%
1071         \noexpand\fi
1072     }%
1073 }%
1074 }
1075 \def\FN@build@symboldef#1{%
1076     \def\@tempa{#1}%
1077     \ifx\@tempa\@tempb
1078     \else
1079         \toks@\expandafter{\the\toks@or#1}%
1080         \expandafter\FN@build@symboldef
1081     \fi
1082 }
1083 \DeclareDocumentCommand\DefineFNsymbolsTM {smm}{%
1084     \expandafter\ifx\csname FN@fnsymbol@#2\endcsname\relax
1085         \PackageInfo{footmisc}{Declaring symbol style #2}%
1086     \else
1087         \PackageWarning{footmisc}{Redefining symbol style #2}%
1088     \fi
1089     \toks@{}%
1090     \def\@tempb{\end}%
1091     \FN@build@symboldefTM#3\end\@null
1092     \expandafter\xdef\csname FN@fnsymbol@#2\endcsname##1{%
1093         \noexpand\ifcase##1%
1094             \the\toks@
1095         \noexpand\else
1096             \IfBooleanTF#1{\noexpand\@ctrerr}%
1097             {\noexpand\FN@orange##1}%
1098         \noexpand\fi
1099     }%
1100 }
1101 \def\FN@build@symboldefTM#1#2{%
1102     \def\@tempa{#1}%
1103     \ifx\@tempa\@tempb
1104     \else
1105         \toks@\expandafter{\the\toks@or\TextOrMath{#1}{#2}}%
1106         \expandafter\FN@build@symboldefTM
1107     \fi
1108 }
1109 \def\FN@orange#1{%
1110     \ifFN@robust
1111         \@arabic#1%
1112         \@bsphack
1113         \PackageInfo{footmisc}{Footnote number \number#1 out of range}%
1114         \protect\@fnsymbol@orange
1115         \@esphack
1116     \else \@ctrerr \fi
1117 }
1118 \global\let\@diagnose@fnsymbol@orange\relax
1119 \AtEndDocument{\@diagnose@fnsymbol@orange}
1120 \def\@fnsymbol@orange{%
1121     \gdef\@diagnose@fnsymbol@orange{%
1122         \PackageWarningNoLine{footmisc}{Some footnote number(s)
1123             were out of range

```

```

1124     \MessageBreak
1125     see log for details%
1126   }%
1127 }%
1128 }
1129 \DefineFNSymbolsTM{bringhurst}{%
1130   \textasteriskcentered *%
1131   \textdagger    \dagger
1132   \textdaggerdbl \ddagger
1133   \textsection   \mathsection
1134   \textbardbl    \|%
1135   \textparagraph \mathparagraph
1136 }%
1137 \DefineFNSymbolsTM{chicago}{%
1138   \textasteriskcentered *%
1139   \textdagger    \dagger
1140   \textdaggerdbl \ddagger
1141   \textsection   \mathsection
1142   \textbardbl    \|%
1143   \#\%#%
1144 }%
1145 \DefineFNSymbolsTM{wiley}{%
1146   \textasteriskcentered *%
1147   {\textasteriskcentered\textasteriskcentered}{**}%
1148   \textdagger    \dagger
1149   \textdaggerdbl \ddagger
1150   \textsection   \mathsection
1151   \textparagraph \mathparagraph
1152   \textbardbl    \|%
1153 }%
1154 \DefineFNSymbolsTM{lamport-robust}{%
1155   \textasteriskcentered *%
1156   \textdagger    \dagger
1157   \textdaggerdbl \ddagger
1158   \textsection   \mathsection
1159   \textparagraph \mathparagraph
1160   \textbardbl    \|%
1161   {\textasteriskcentered\textasteriskcentered}{**}%
1162   {\textdagger\textdagger}{\dagger\dagger}%
1163   {\textdaggerdbl\textdaggerdbl}{\ddagger\ddagger}%
1164 }
1165 \DefineFNSymbolsTM*{lamport*}{%
1166   \textasteriskcentered *%
1167   \textdagger    \dagger
1168   \textdaggerdbl \ddagger
1169   \textsection   \mathsection
1170   \textparagraph \mathparagraph
1171   \textbardbl    \|%
1172   {\textasteriskcentered\textasteriskcentered}{**}%
1173   {\textdagger\textdagger}{\dagger\dagger}%
1174   {\textdaggerdbl\textdaggerdbl}{\ddagger\ddagger}%
1175   {\textsection\textsection}{\mathsection\mathsection}%
1176   {\textparagraph\textparagraph}{\mathparagraph\mathparagraph}%
1177   {\textasteriskcentered\textasteriskcentered\textasteriskcentered}{***}%

```

```

1178 {\textdagger\textdagger\textdagger}{\dagger\dagger\dagger}%
1179 {\textdaggerdbl\textdaggerdbl\textdaggerdbl}{\ddagger\ddagger\ddagger}%
1180 {\textsection\textsection\textsection}%%
1181 {\mathsection\mathsection\mathsection}%
1182 {\textparagraph\textparagraph\textparagraph}%%
1183 {\mathparagraph\mathparagraph\mathparagraph}%
1184 }
1185 \setfnsymbol{lamport*}
1186 \DefineFNSymbolsTM{lamport*--robust}{%
1187   \textasteriskcentered *%
1188   \textdagger \dagger
1189   \textdaggerdbl \ddagger
1190   \textsection \mathsection
1191   \textparagraph \mathparagraph
1192   \textbardbl \l%
1193   {\textasteriskcentered\textasteriskcentered}{**}%
1194   {\textdagger\textdagger}{\dagger\dagger}%
1195   {\textdaggerdbl\textdaggerdbl}{\ddagger\ddagger}%
1196   {\textsection\textsection}{\mathsection\mathsection}%
1197   {\textparagraph\textparagraph}{\mathparagraph\mathparagraph}%
1198   {\textasteriskcentered\textasteriskcentered\textasteriskcentered}{***}%
1199   {\textdagger\textdagger\textdagger}{\dagger\dagger\dagger}%
1200   {\textdaggerdbl\textdaggerdbl\textdaggerdbl}{\ddagger\ddagger\ddagger}%
1201   {\textsection\textsection\textsection}%%
1202   {\mathsection\mathsection\mathsection}%
1203   {\textparagraph\textparagraph\textparagraph}%%
1204   {\mathparagraph\mathparagraph\mathparagraph}%
1205 }
1206 \newcommand\mpfootnotemark{%
1207   \@ifnextchar[%]
1208     \@xmpfootnotemark
1209     {%
1210       \stepcounter\@mpfn
1211       \protected@xdef\@thefnmark{\thempfn}%
1212       \@footnotemark
1213     }%
1214 }
1215 \def\@xmpfootnotemark[#1]{%
1216   \begingroup
1217     \csname c@\@mpfn\endcsname #1\relax
1218     \unrestored@protected@xdef\@thefnmark{\thempfn}%
1219   \endgroup
1220   \@footnotemark
1221 }
1222 \endinput
1223 </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.

Symbols	
<code>\#</code>	1143
<code>\ </code>	1134, 1142, 1152, 1160, 1171, 1192
A	
<code>\addtocounter</code>	890
<code>\AddToHook</code>	319, 397, 452, 454, 460, 462, 482, 682, 814, 818, 850, 903, 911, 981, 982
<code>\AddToHookWithArguments</code>	481
<code>\advance</code>	658, 732, 929
<code>\AssignSocketPlug</code> ..	209, 292, 304, 307, 316, 353, 551, 594, 595, 609, 618, 686, 687, 688, 689, 761, 787, 789, 793, 795, 799, 801, 830, 831, 832, 833
<code>\AtBeginDocument</code>	326, 464, 469
<code>\AtEndDocument</code>	1119
<code>\AtEndOfPackage</code>	667
B	
<code>\baselineskip</code>	728, 859
<code>\baselinestretch</code>	462, 814, 818
<code>\begingroup</code>	437, 1216
<code>\bgroup</code>	917
bool commands:	
<code>\bool_gset_false:N</code>	118
<code>\bool_gset_true:N</code>	117
<code>\bool_if:NTF</code> ...	121, 133, 521, 537, 577
<code>\bool_new:N</code>	69, 76, 116
<code>\bool_set_false:N</code> ..	207, 440, 473, 476
<code>\bool_set_true:N</code>	70, 77, 443
<code>\box</code>	302
C	
<code>\columnwidth</code> ..	16, 238, 301, 878, 879, 928
<code>\cs</code>	247, 248
cs commands:	
<code>\cs_generate_variant:Nn</code>	80, 81, 502, 516, 561
<code>\cs_gset_eq:NN</code>	128, 152
<code>\cs_if_eq:NNTF</code>	399, 404
<code>\cs_new:Npn</code>	388, 390, 392, 394, 396
<code>\cs_new_eq:NN</code>	197, 362
<code>\cs_new_protected:Npn</code>	82, 95, 117, 118, 119, 131, 155, 182, 211, 332, 364, 480, 489, 495, 503, 552, 556
<code>\cs_set:Npn</code>	385
<code>\cs_set_eq:NN</code> 199, 321, 368, 371, 374, 377, 379, 401, 424, 426, 456, 457, 458	
<code>\cs_set_protected:Npn</code>	484
<code>\csname</code> 20, 884, 1011, 1012, 1026, 1027, 1041, 1049, 1059, 1084, 1092, 1217	
<code>\CurrentOption</code>	671, 673, 678, 680
D	
<code>\dagger</code> ..	1131, 1139, 1148, 1156, 1162, 1167, 1173, 1178, 1188, 1194, 1199
<code>\ddagger</code> ..	1132, 1140, 1149, 1157, 1163, 1168, 1174, 1179, 1189, 1195, 1200
<code>\DebugFNotesOff</code>	7, <u>117</u>
<code>\DebugFNotesOn</code>	7, <u>117</u>
<code>\DeclareCurrentRelease</code>	642, 645
<code>\DeclareDocumentCommand</code>	416, 421, 430, 435, 1048, 1083
<code>\DeclareOption</code>	662, 664, 670, 677, 693, 700, 707, 712, 716, 717, 718, 721, 725, 730, 734, 753, 755
<code>\DeclareRelease</code>	641, 644
<code>\def</code> 12, 18, 29, 30, 34, 242, 385, 463, 656, 657, 738, 739, 764, 856, 863, 875, 879, 884, 888, 892, 916, 955, 984, 988, 1004, 1012, 1017, 1019, 1027, 1032, 1055, 1057, 1058, 1075, 1076, 1090, 1101, 1102, 1109, 1120, 1215	
default (plug) 5, 5, 5, 5, 5, 5, 5, 5, 5, 6, 22	
<code>\DefineFNSymbols</code>	1048
<code>\DefineFNSymbolsTM</code>	1083, 1129, 1137, 1145, 1154, 1165, 1186
<code>\dimexpr</code>	300
<code>\dp</code>	236, 298
E	
<code>\edef</code>	164, 656, 991, 1010, 1025
<code>\egroup</code>	947
<code>\else</code> ..	341, 344, 347, 681, 742, 745, 765, 772, 779, 788, 794, 800, 803, 866, 878, 904, 913, 922, 939, 951, 960, 963, 999, 1006, 1021, 1051, 1062, 1068, 1078, 1086, 1095, 1104, 1116
<code>\end</code>	1055, 1056, 1090, 1091
<code>\endcsname</code>	20, 884, 1011, 1012, 1026, 1027, 1042, 1049, 1059, 1084, 1092, 1217
<code>\endgroup</code>	439, 1219
<code>\endinput</code>	1222
<code>\ensuremath</code>	1061
<code>\ERROR</code>	23, 362, 378, 414, 784, 804

exp commands:

- \exp_after:wN 384, 386
- \exp_args:No 539
- \exp_not:N 529
- \exp_stop_f: ... 379, 389, 391, 393, 395
- \expandafter
 - 657, 1005, 1007, 1011, 1012, 1020,
 - 1022, 1026, 1027, 1041, 1049, 1059,
 - 1079, 1080, 1084, 1092, 1105, 1106
- \ExplSyntaxOff 619, 980
- \ExplSyntaxOn 64, 978

F

FEMark (plug) 517

FENote (plug) 562

FENoteLbl (plug) 596

FENotetext (plug) 610

\fi 173, 191, 330,

349, 350, 351, 690, 747, 749, 762,

774, 776, 782, 790, 796, 802, 805,

835, 869, 871, 878, 890, 893, 907,

924, 941, 965, 966, 970, 971, 997,

1001, 1008, 1023, 1033, 1053, 1064,

1071, 1081, 1088, 1098, 1107, 1116

\floatingpenalty 237

fnmark (hook) 4, 4, 107

fnmark/after (hook) 4, 4, 4, 6, 17, 107

fnmark/before (hook) 4, 4, 4, 107

fnmark/begin (hook) 4, 4, 107

fnmark/end (hook) 4, 4, 17, 107

fnote commands:

- \fnote_class_new:nn .. 9, 489, 489, 494
- \g_fnote_debug_bool
 - 116, 117, 118, 121, 133
- \fnote_footnotemark:
 - 8, 155, 155, 199, 456
- \fnote_footnotetext:n
 - 200, 211, 321, 461
- \fnote_gput_refs:nn
 - 529, 552, 556, 561, 582
- \l_fnote_link_bool 27, 69, 473, 476, 537
- \l_fnote_link_type_tl 71, 541
- \fnote_makefntext:n
 - 11, 22, 24, 332, 332, 401
- \fnote_mark_gpop_all:nnN
 - 9, 503, 503, 516, 579
- \fnote_mark_gput:nn
 - 9, 495, 495, 502, 523
- \fnote_set_fnmark:nn 95, 95, 418
- \fnote_step_fnmark:nn 82, 82, 423, 432
- \l_fnote_type_tl 67, 523, 579

fnote internal commands:

- \l__fnote_autodetect_bool
 - 76, 440, 443, 521, 577
- \l__fnote_currentlabel_tl
 - 25, 29, 74, 441, 531, 533
- \l__fnote_currentrefs_seq
 - 75, 579, 580, 599, 600
- \l__fnote_currentstruct_tl
 - 27, 78, 79, 479, 576, 582, 601, 602
- __fnote_debug_footnotemark: ...
 - 119, 128, 156
- __fnote_debug_footnotetext: ...
 - 131, 152, 212
- __fnote_footmark_finish: 17
- __fnote_footnotemark_default_-
 - finish: 17, 182, 182, 197, 426
- __fnote_footnotemark_finish: ..
 - 17, 180, 182, 197, 424, 426, 428
- __fnote_gput_ref:nn 552, 552, 558, 559
- __fnote_label_hook:n
 - 480, 480, 481, 484
- \l__fnote_linktarget_tl
 - 73, 524, 533, 542
- __fnote_patch: 364, 405
- __fnote_patch_@makefnmark:w 377, 394
- __fnote_patch_hb@xt:w ... 371, 390
- __fnote_patch_hbox:w 368, 388
- __fnote_patch_makebox:w .. 374, 392
- \l__fnote_patch_seq 413, 414
- \l__fnote_patch_tl 361, 366,

367, 370, 373, 376, 383, 384, 386, 413
- __fnote_tmp:w 362,

368, 371, 374, 377, 379, 384, 385, 386
- \l__fnote_tmpa_prop 65, 506, 507
- \l__fnote_tmpa_tl 66
- \fnsymbol 662
- fntext (hook) 6, 6, 111
- fntext/after (hook) 6, 6, 6, 111
- fntext/before (hook) 6, 6, 6, 111
- fntext/begin (hook) 6, 6, 6, 111
- fntext/begin (socket) 5, 5, 305
- fntext/end (hook) 6, 6, 6, 111
- fntext/end (socket) 5, 5, 308
- fntext/make (socket) ... 5, 5, 5, 5, 6, 293
- fntext/mark (socket) 5, 5, 337
- fntext/para (hook) 6, 6, 6, 6, 22, 111
- fntext/process (socket) 5, 5, 289
- fntext/text (socket) 6, 6, 354
- \footglue 314, 843
- \footins 5, 290, 732, 759, 860
- \footnote 5, 9, 11, 20, 421, 1003, 1004
- \footnotebaselineskip . 301, 846, 853, 859
- \footnotelayout ... 692, 695, 696, 945, 967
- \footnotemargin 5, 7, 21, 323, 324,

328, 329, 339, 340, 342, 345, 348,

675, 719, 722, 899, 904, 905, 906,

920, 921, 937, 938, 958, 959, 961, 964

<code>\footnotemark</code>	4, 9–11, 430, 1018, 1019	<code>\ifx</code>	740, 765, 1004, 1019, 1049, 1060, 1077, 1084, 1103
<code>\footnoterule</code>	731, 736, 737, 739	<code>\ignorespaces</code>	23, 270
<code>\footnotesep</code>	23, 235, 306, 684, 868	<code>\insert</code>	5, 8, 17, 290
<code>\footnotesize</code>	15, 228, 852, 931	<code>\insertpenalties</code>	748
<code>\footnotetext</code>	5, 6, 9–11, 20, 416	int commands:	
<code>\footref</code>	10, 11, 25, 27, 32, 435	<code>\int_compare:nTF</code>	414
G		<code>\int_max:nn</code>	573
<code>\gdef</code>	735, 1121	<code>\int_set:Nn</code>	90, 102
<code>\global</code>	13, 447, 759, 773, 853, 884, 1118	<code>\interfootnotelinepenalty</code>	234
<code>\gluestretchorder</code>	768	<code>\interlinepenalty</code>	6, 234
group commands:		K	
<code>\group_begin:</code>	89, 101	<code>\kern</code>	738, 985, 986
<code>\group_end:</code>	92, 104	L	
H		<code>\label</code>	10
<code>\hangfootparindent</code>	729, 944	<code>\lastbox</code>	863
<code>\hangfootparskip</code>	728, 943	<code>\lastkern</code>	989
<code>\hbox</code>	23, 24, 34, 297, 367, 388, 389, 410, 413, 919	<code>\lastskip</code>	767
hook commands:		<code>\leavevmode</code>	162, 934
<code>\hook_gput_code:nnn</code>	80, 527	<code>\leftmargin</code>	926, 929, 930
<code>\hook_log:n</code>	123, 124, 125, 126, 127, 146, 147, 148, 149, 150, 151	legacy commands:	
<code>\hook_use:n</code>	160, 170, 176, 215, 225, 251, 268, 275, 286	<code>\legacy_if:nTF</code>	473
Hooks:		<code>\let</code>	453, 461, 462, 661, 692, 705, 708, 713, 736, 737, 773, 778, 780, 781, 814, 815, 818, 819, 840, 949, 1000, 1003, 1018, 1041, 1046, 1118
<code>fnmark</code>	4, 4, 107	<code>\linepenalty</code>	695, 696
<code>fnmark/after</code>	4, 4, 4, 6, 17, 107	<code>\linewidth</code>	928, 929, 930
<code>fnmark/before</code>	4, 4, 4, 107	<code>\llap</code>	5, 7, 343, 348, 936, 962, 964
<code>fnmark/begin</code>	4, 4, 107	<code>\long</code>	12, 385, 856, 916, 955
<code>fnmark/end</code>	4, 4, 17, 107	<code>\ltxlabfootnotedate</code>	4
<code>fnntext</code>	6, 6, 111	<code>\ltxlabfootnoteversion</code>	4
<code>fnntext/after</code>	6, 6, 6, 111	M	
<code>fnntext/before</code>	6, 6, 6, 111	<code>\makebox</code>	24, 373, 392, 393
<code>fnntext/begin</code>	6, 6, 6, 111	<code>\MakeLinkTarget</code>	600, 601
<code>fnntext/end</code>	6, 6, 6, 111	<code>\MakePerPage</code>	38, 883
<code>fnntext/para</code>	6, 6, 6, 6, 22, 111	<code>\marginpar</code>	291
<code>\hrule</code>	738	<code>\marginparwidth</code>	683
<code>\hsize</code>	6, 19, 16, 238, 683, 876, 879	<code>\mathparagraph</code>	1135, 1151, 1159, 1170, 1176, 1183, 1191, 1197, 1204
<code>\hskip</code>	314	<code>\mathsection</code>	1133, 1141, 1150, 1158, 1169, 1175, 1181, 1190, 1196, 1202
<code>\hspace</code>	911	<code>\maxdimen</code>	5, 7, 324, 328, 345, 675, 900
<code>\hss</code>	340, 348, 360, 921, 938, 959, 964	<code>\MessageBreak</code>	1124
<code>\ht</code>	299	<code>mp (plug)</code>	5
I		<code>\mpfootnotemark</code>	1206
<code>identity (plug)</code>	6, 17, 18, 22	<code>\mpfootnoterule</code>	737, 741
<code>\IfBooleanTF</code>	1069, 1096	<code>\multfootsep</code>	979, 995
<code>\ifcase</code>	783, 1066, 1093	<code>\multiplefootnotemarker</code>	977, 985, 986, 989
<code>\ifdim</code>	328, 339, 342, 345, 904, 920, 937, 958, 961, 989	N	
<code>\ifhbox</code>	864	<code>\NeedsTeXFormat</code>	640, 651
<code>\ifhmode</code>	163, 189		
<code>\ifnum</code>	743, 768, 876, 889		

Sockets:		
fntext/begin	5, 5, 305	
fntext/end	5, 5, 308	
fntext/make	5, 5, 5, 5, 5, 6, 293	
fntext/mark	5, 5, 337	
fntext/process	5, 5, 289	
fntext/text	6, 6, 354	
tagsupport/fnmark	8, 8, 10, 29, 198	
tagsupport/fntext/begin	8, 8, 11, 317	
tagsupport/fntext/end	8, 8, 11, 317	
tagsupport/fntext/mark	8, 8, 11, 32, 355	
tagsupport/fntext/text	8, 8, 11, 32, 355	
\space	4	
\spacefactor	164, 190, 991, 996	
\splitfootnoterule	738, 746	
\splitmaxdepth	236	
\splittopskip	235	
\stepcounter	38, 85, 885, 892, 1210	
\strut	312	
\strutbox	23, 236, 309	
\SuspendTagging	918	
T		
tag commands:		
\tag_check_child:nnTF	565	
\tag_get:n	498, 524, 530, 573, 576	
\tag_mc_begin:n	535, 603, 613, 979	
\tag_mc_begin_pop:n	549, 592, 607, 616, 979	
\tag_mc_end:	547, 605, 615, 979	
\tag_mc_end_push:	519, 564, 598, 612, 979	
\tag_struct_begin:n	520, 567, 570, 602	
\tag_struct_end:	548, 591, 606	
\tag_struct_gput:nnn	554	
\tag_struct_use:n	81	
tag internal commands:		
\l__tag_para_flattened_bool	207	
\l__tag_para_main_tag_tl	205	
\l__tag_para_tag_default_tl	206	
\l__tag_para_tag_tl	206	
tagsupport/fnmark (socket)	8, 8, 10, 29, 198	
tagsupport/fntext/begin (socket)	8, 8, 11, 317	
tagsupport/fntext/end (socket)	8, 8, 11, 317	
tagsupport/fntext/mark (socket)	8, 8, 11, 32, 355	
tagsupport/fntext/text (socket)	8, 8, 11, 32, 355	
T _E X and L ^A T _E X 2 _ε commands:		
\@@dvance@macro	656, 657	
\@@par	932	
\@MM	237	
\@advance@macro	661	
\@arabic	1111	
\@auxout	654	
\@bsphack	1035, 1112	
\@ctrerr	1069, 1096, 1116	
\@currentcounter	18, 242	
\@currentlabel	19, 243	
\@diagnose@fnsymbol@orange	1118, 1119, 1121	
\@eha	1039	
\@empty	467, 692	
\@esphack	1044, 1115	
\@finalstrut	23, 309	
\@fnsymbol	665, 1041, 1046	
\@fnsymbol@orange	1114, 1120	
\@footnotemark	3, 8, 16, 25, 199, 425, 433, 442, 456, 1212, 1220	
\@footnotetext	3, 18, 321, 419, 427, 461	
\@gobble	1015	
\@gobbletwo	466	
\@ifclassloaded	812	
\@ifnextchar	1013, 1028, 1207	
\@ifpackageloaded	471, 809, 874	
\@ifundefined	694, 1036	
\@listdepth	740	
\@makecol	453, 458	
\@makefnmark	3–5, 7, 8, 10, 17, 23, 24, 34, 178, 340, 343, 346, 348, 360, 376, 394, 395, 921, 923, 938, 940, 959, 962, 964	
\@makefntext	3, 5, 6, 11, 21–24, 29, 22, 241, 255, 294, 297, 366, 385, 399, 401, 404, 457, 949, 955	
\@makefntext@processX	389, 391, 393, 395, 396	
\@mpfn	29, 418, 423, 1210, 1217	
\@mpfootins	13, 14, 447, 448	
\@mpfootnotetext	9, 12	
\@mplistdepth	740	
\@one	708, 876, 930	
\@null	1091	
\@outputbox@append	766, 771	
\@outputbox@reinsertbskip	770, 773, 778, 781	
\@outputbox@removebskip	764, 780	
\@parboxrestore	17, 239, 858	
\@plus	732	
\@setpar	932	
\@stpelt	884	
\@tempa	1076, 1077, 1102, 1103	
\@tempb	1055, 1077, 1090, 1103	
\@tempboxa	840, 919, 926	
\@tempc	1057, 1060	
\@tempcnta	657, 658, 659	
\@tempd	1058, 1060	

\@tempskipa	767, 768, 769, 771	\footmisc@hang@makefntext	404, 916, 949
\@textbottom	765	\hb@xt@	23, 24, 340, 348, 360, 370, 390, 391, 921, 938, 959, 964
\@textsuperscript	34	\hyper@linkend	467, 544
\@thefnmark	9, 10, 13, 14, 25, 28, 29, 31, 20, 34, 86, 91, 98, 103, 243, 438, 523, 579, 1211, 1218	\hyper@linkstart	466, 540
\@typeset@protect	1004, 1019	\hyper@nopatch@footnote	463
\@x@sf	164, 190, 991, 996	\if@tempswb	1047
\@xmpfootnotemark	1208, 1215	\ifFN@abovefloats	706, 786, 792, 798
\c@footnote	665	\ifFN@fixskip	703, 763
\color@begingroup	21, 253	\ifFN@hangfoot	724, 915
\color@endgroup	25, 280	\ifFN@multiplefootnote	754, 976
\doublecol@number	876	\ifFN@para	669, 677, 757, 828, 839, 896
\FN@abovefloatsfalse	709, 717	\ifFN@perpage	698, 881
\FN@abovefloatstrue	706, 714, 716	\ifFN@robust	663, 1110
\FN@baselinestretch	815, 819	\ifFN@setspace	808
\FN@bottomcases	705, 708, 713, 783	\ifFN@stablefootnote	752, 1002
\FN@build@symboldef	1056, 1075, 1080	\l@advance@macro	656, 661
\FN@build@symboldefTM	1091, 1101, 1106	\m@m@singlespace	814, 815
\FN@fixskipfalse	703	\m@ne	884, 890
\FN@fixskiptrue	710, 714, 716, 717	\new@std@class@makecol	453, 458
\FN@fnsymbol@lamport	1046	\nfss@text	1063
\FN@hangfootfalse	724	\old@std@class@makefntext	357, 399, 457
\FN@hangfoottrue	726	\p@	732, 738
\FN@makefootnoteparagraph	759, 856	\p@footnote	243
\FN@m@check	981, 988	\pp@cl@end@iii	892
\FN@m@prepare	982, 984, 1000	\pp@fix@MakePerPage	886, 888, 892
\FN@multiplefootnotefalse	754	\protected@edef	19, 243
\FN@multiplefootnotetrue	755	\protected@write	654
\FN@orange	1070, 1097, 1109	\protected@writeaux	653
\FN@parafalse	669	\protected@xdef	86, 98, 1211
\FN@paratrue	674	\reset@font	15, 227
\FN@perpagefalse	699	\setspace@singlespace	462, 818, 819
\FN@perpagetrue	701	\split@prev	735, 743, 748
\FN@removehboxes	860, 863, 864	\thr@@	705
\FN@robustfalse	663	\toks@	1054, 1067, 1079, 1089, 1094, 1105
\FN@robusttrue	666	\tw@	713
\FN@setfootnoteparawidth	857, 875, 879	\unrestored@protected@xdef	91, 103, 438, 1218
\FN@setspacefalse	823	\z@	23, 298, 306, 339, 342, 684, 743, 768, 868, 889, 904, 920, 927, 937, 958, 961
\FN@setspacetrue	811	\textasteriskcentered	1130, 1138, 1146, 1147, 1155, 1161, 1166, 1172, 1177, 1187, 1193, 1198
\FN@sf@@footnote	1003, 1005	\textbardbl	1134, 1142, 1152, 1160, 1171, 1192
\FN@sf@@footnotemark	1018, 1020	\textdagger	1131, 1139, 1148, 1156, 1162, 1167, 1173, 1178, 1188, 1194, 1199
\FN@sf@gobble@bracket	1029, 1032	\textdaggerdbl	1132, 1140, 1149, 1157, 1163, 1168, 1174, 1179, 1189, 1195, 1200
\FN@sf@gobble@opt	1007, 1010	\TextOrMath	1105
\FN@sf@gobble@optonly	1022, 1025		
\FN@sf@gobble@twobracket	1014, 1017		
\FN@stablefootnotefalse	752		
\FN@stablefootnotetrue	753		
\FN@tempboxa	297, 298, 299, 300, 302, 840, 863, 864, 865		
\FN@tempboxb	841		
\FN@tempboxc	842		
\FN@temptoken	652		

<code>\textparagraph</code>	1135, 1151, 1159, 1170, 1176, 1182, 1191, 1197, 1203	<code>\unkern</code>	993, 994
<code>\textsection</code>	1133, 1141, 1150, 1158, 1169, 1175, 1180, 1190, 1196, 1201	<code>\unskip</code>	159
<code>\textsuperscript</code>	995	<code>\unvbox</code>	14, 448, 860
<code>\textwidth</code>	877	use commands:	
<code>\the</code>	164, 659, 748, 771, 991, 1067, 1079, 1094, 1105	<code>\use:N</code>	86, 91, 98, 103
<code>\thefootnote</code>	30, 662, 665	<code>\use:n</code>	395
<code>\thempfn</code>	30, 1211, 1218	<code>\UseHook</code>	187, 194
tl commands:		<code>\UseSocket</code>	366, 396, 935
<code>\tl_if_eq:nnTF</code>	509	<code>\UseTaggingSocket</code>	8, 240
<code>\tl_if_in:NnTF</code>	367, 370, 373, 376		
<code>\tl_if_novalue:nTF</code>	83, 96	V	
<code>\tl_new:N</code>	66, 67, 71, 73, 74, 78, 361	<code>\value</code>	889
<code>\tl_set:Nn</code>	68, 72, 79, 205, 366, 383, 441, 524, 533, 576	<code>\vbox</code>	13, 447, 759
<code>\tl_set_eq:NN</code>	206	<code>\vskip</code>	769, 771
		W	
U		<code>\wd</code>	300, 926
<code>\unhbox</code>	865	X	
		<code>\xdef</code>	748, 770, 1059, 1092