

The latex-lab-footnotes code*

Frank Mittelbach, L^AT_EX Project

July 31, 2023

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
3	Tagging and hyperlinking support	7
3.1	Technical details for the tagging	7
3.2	Requirements for links	8
3.3	The algorithmus to connect marks and notes	8
3.3.1	\footref	9
3.3.2	\footnotemark after \footnotetext	9
3.4	Links	10
3.5	Implementation details regarding tagging	10
3.6	Handling the mark	10
3.7	Handling the footnotetext	10
3.8	Footnotes in minipages	10
4	TODOs	10

*

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

1 Introduction

This code reimplements the footnote interfaces for L^AT_EX 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.

3 Tagging and hyperlinking support

TODO: *this section needs work (and probably csname 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:

tagsupport/fnmark (1 argument)

TODO: *describe and decide on names*

tagsupport/fntext/begin (no argument)

tagsupport/fntext/end (no argument)

tagsupport/fntext/mark (1 argument)

`tagsupport/fntext/text` (1 argument)

The *footnotemark* should create a `/Lb1` 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 want to avoid faulty nesting.
- It must detect which marks are related to which notes so that it can setup the `/Ref` cross-references.

3.2 Requirements for links

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

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

It should be possible to suppress the links both globally and locally.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 a match it stores the structure numbers and removes the entries from the properties (so in a normal document the property will never contain more than a few entries).

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

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`.

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.

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`

<hr/> <code>\fnote_class_new:nn</code> <hr/>	<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.
<hr/> <code>\fnote_mark_gput:nn</code> <code>\fnote_mark_gput:(no oo)</code> <hr/>	<code>\fnote_mark_gput:nn{<mark>}{<class name>}</code>	This command stores the current structure number as key and the <code><mark></code> as value in the property list associated with the <code><class name></code> .
<hr/> <code>\fnote_mark_gpop_all:nnN</code> <hr/>	<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 <code><class name></code> are equal to <code><mark></code> into the sequence <code><sequence></code> and then removes them from the property list. The content of the sequence can then be used to create link targets and references.

3.3.1 `\footref`

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

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

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

3.3.2 `\footnotemark` after `\footnotetext`

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

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

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

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

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

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

3.4 Links

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

A `\footnotetext` creates a bunch of destinations (in most cases this sums up to two destinations): one for every structure number in the `/Ref` (used as target by the mark commands) and one for the structure number of the `footnotetext` itself (used as target by `\footref` 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 $\text{\LaTeX 2}_{\epsilon}$ 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

- tagging destroys footnotes directly following the text with `pdflatex`.
- there is a dependency to `etoolbox` as we patch `\iiiminipage`

- 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 `\@makefnmark`.

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           [\lftlabfootnotedate\space v\lftlabfootnoteversion\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 %       }
25 %     \color@endgroup}}
26 % =====
27 % used by the minipage footnote code.
28 %
29 % \def\mpfn{footnote}
30 % \def\thempfn{\thefootnote}
31 % =====
32 % this perhaps need some configuration options.
33 %
```

```

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

```

```

64 \ExplSyntaxOn

```

5.3 Temporary variables

```

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

```

5.4 Public variables

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

```

\l_fnote_type_tl

```

```

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

```

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

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

`\l_fnote_link_bool`

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

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

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

`\l_fnote_link_type_tl`

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

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

5.5 Internal variables

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

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

(End of definition for \l__fnote_linktarget_tl.)

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

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

(End of definition for \l__fnote_currentlabel_tl.)

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

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

(End of definition for \l__fnote_currentrefs_seq.)

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

`\l__fnote_autodetect_bool`

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

(End of definition for \l__fnote_autodetect_bool.)

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

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

5.6 Variants

```

80 \cs_generate_variant:Nn \ref_value:nn { Vn }
81 \cs_generate_variant:Nn \prop_gput:Nnn { cxn }
82 \cs_generate_variant:Nn \hook_gput_code:nnn{nne}
83 \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.

```

84 \cs_new_protected:Npn \fnote_step_fnmark:nn #1#2 {
85   \tl_if_novalue:nTF {#1}
86   {
87     \stepcounter {#2}
88     \protected@xdef \@thefnmark { \use:c { the#2 } }
89   }
90   {
91     \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.

```

92       \int_set:cn { c@#2 }{ #1 }
93       \unrestored@protected@xdef \@thefnmark { \use:c { the#2 } }
94     \group_end:
95   }
96 }

```

(End of definition for \fnote_step_fnmark:nn. This function is documented on page ??.)

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

```

97 \cs_new_protected:Npn \fnote_set_fnmark:nn #1#2 {
98   \tl_if_novalue:nTF {#1}
99   {
100     \protected@xdef \@thefnmark { \use:c { the#2 } }
101   }
102   {
103     \group_begin:
104       \int_set:cn { c@#2 }{ #1 }
105       \unrestored@protected@xdef \@thefnmark { \use:c { the#2 } }
106     \group_end:
107   }
108 }

```

(End of definition for \fnote_set_fnmark:nn. This function is documented on page ??.)

5.8 Hooks

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

`fnmark/after` (*hook*) `\NewMirroredHookPair{fnmark/before}{fnmark/after}`

`fnmark` (*hook*) `\NewHook{fnmark}`

`fnmark/begin` (*hook*) `\NewHook{fnmark/begin}`

`fnmark/end` (*hook*) `\NewHook{fnmark/end}`

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

```

fntext/after (hook) 113 \NewMirroredHookPair{fntext/before}{fntext/after}
      fntext (hook) 114 \NewHook{fntext}
fntext/begin (hook) 115 \NewHook{fntext/para}
      fntext/end (hook) 116 \NewHook{fntext/begin}
fntext/para (hook) 117 \NewHook{fntext/end}

```

5.9 Debugging code

the debugging code is just temporary

For now we have debugging turned on by default

```

118 \bool_new:N          \g_fnote_debug_bool
119 \bool_gset_true:N    \g_fnote_debug_bool

```

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

```

120 \cs_new_protected:Npn \__fnote_debug_footnotemark:
121 {
122   \bool_if:NT \g_fnote_debug_bool
123   {
124     \hook_log:n {fnmark/before}
125     \hook_log:n {fnmark}
126     \hook_log:n {fnmark/begin}
127     \hook_log:n {fnmark/end}
128     \hook_log:n {fnmark/after}
129     \cs_gset_eq:NN \__fnote_debug_footnotemark: \prg_do_nothing:
130   }
131 }

```

Similar for the `footnotetext`

```

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

```

Show the info only once (if at all).

```

153         \cs_gset_eq:NN \__fnote_debug_footnotetext: \prg_do_nothing:
154     }
155 }

```

5.10 The new \@footnotemark command

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

```

156 \cs_new_protected:Npn \fnote_footnotemark: {
157     \__fnote_debug_footnotemark:
158     %-----
159     % bibarts
160     % chextris --- actually in the wrong place does an \unskip
161     \hook_use:n {fnmark/before}
162     %-----
163     \leavevmode
164     \ifhmode
165         \edef\@x@sf{\the\spacefactor}
166     %-----
167     % bxjsja-minimal.def --- what they do could be done at ‘‘bibarts’’
168     %                                     (a bit less efficient)
169     % memhfixc.sty
170     % footmisc.sty
171     \hook_use:n {fnmark}
172     %-----
173     \nobreak
174     \fi
175     %-----
176     % hyperref.sty
177     \hook_use:n {fnmark/begin}
178     %-----

```

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

```

179     \socket_use:nn {taggsupport/fnmark} \@makefnmark
180     %-----

```

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

```

181     \__fnote_footnotemark_finish:
182 }

```

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

`__fnote_footnotemark_default_finish:` The default definition for `__fnote_footnotemark_finish:` is called `__fnote_footnotemark_default_finish:`

```

183 \cs_new_protected:Npn \__fnote_footnotemark_default_finish: {
184 % hyperref.sty
185 % memhfixc.sty --- could move fnmark/after
186 % scrllttr2.cls --- could vanish if footmisc uses a hook
187 % footmisc.sty

```



```

188 \UseHook{fnmark/end}
189 %-----
190 \ifhmode
191 \spacefactor \@x@sf \relax
192 \fi
193 %
194 %-----
195 \UseHook{fnmark/after}
196 %-----
197 }

```

```

198 \cs_new_eq:NN \__fnote_footnotemark_finish: \__fnote_footnotemark_default_finish:

```

(End of definition for __fnote_footnotemark_default_finish: and __fnote_footnotemark_finish:.)

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

```

199 \NewSocket{tagsupport/fnmark}{1}

```

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

```

200 \cs_set_eq:NN \@footnotemark \fnote_footnotemark:

```

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

5.11 The new \fnote_footnotetext command

`\fnote_footnotetext:n`

```

201 \cs_new_protected:Npn \fnote_footnotetext:n #1 {
202 \__fnote_debug_footnotetext:
203 %-----
204 % ./linguex/linguex.sty
205 \hook_use:n {fntext/before}
206 %-----

```

Execute a kernel socket for tagging.

```

207 \socket_use:n {tagsupport/fntext/begin}
208 \socket_use:nn {fntext/process}
209 {
210 %-----
211 % resetting baselinestretch ... (could be done further down)
212 % ./uafthesis/uafthesis.cls
213 % ./setspace/setspace.sty
214 % ./footmisc/footmisc.sty (normal)
215 \hook_use:n {fntext}
216 %-----
217 \reset@font
218 \footnotesize
219 %-----
220 % some classes use a different font size, e.g.,
221 % ./nrc/nrc1.cls ./nrc/nrc2.cls
222 % but those could be done in fntext/para instead
223 %-----

```

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

```

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

```

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

```

266 \socket_use:n {fntext/end}
267 }
268 \par
269 \color@endgroup
270 }
271 %-----

```

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

```

272 \socket_use:n{taggsupport/fntext/end}
273 %-----
274 % ./linguex/linguex.sty
275 \hook_use:n {fntext/after}
276 %-----
277 }

```

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

fntext/process (socket)

```

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

```

fntext/make (socket) This socket receives the $\langle text \rangle$ from the \footnote or \footnotetext and formats it.

```

282 \NewSocket      {fntext/make}{1}
283 \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 T_EXbook algorithm in appendix D).

```

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

```

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

```

294 \NewSocket      {fntext/begin}{0}
295 \NewSocketPlug{fntext/begin}{default}{ \rule\z@\footnotesep }
296 \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.

```

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

```

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

```

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

```

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

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

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

```
308 \AddToHook{begindocument}  
309 {  
310   \cs_set_eq:NN \@footnotetext \fnote_footnotetext:n  
311 }
```

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.

```
312 \newdimen\footnotemargin  
313 \footnotemargin\maxdimen % no value given  
314  
315 \AtBeginDocument  
316 {  
317   \ifdim \footnotemargin=\maxdimen  
318     \setlength\footnotemargin{1.8em}  
319   \fi  
320 }
```

`\fnote_makefntext:n`

```
321 \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.

```
322 \noindent  
323 \socket_use:nn {tagsupport/fntext/mark} { \socket_use:n {fntext/mark} }  
324 \socket_use:nn {tagsupport/fntext/text} { \socket_use:nn {fntext/text}{#1} }  
325 }
```

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

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

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

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

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

```

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

```

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

```

343 \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.

```

344 \NewSocket{tagssupport/fntext/mark}{1}
345 \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
```

```

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

```

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

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

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

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

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

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

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

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

```

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

```

```

352
353 \cs_new_protected:Npn \__fnote_patch:
354 {
355   \tl_set:No \l__fnote_patch_tl { \@makefntext { \UseSocket{tagsupport/fntext/text}{##1} } }
356   \tl_if_in:NnTF \l__fnote_patch_tl { \hbox }
357   { \cs_set_eq:NN \__fnote_tmp:w \__fnote_patch_hbox:w }
358   {
359     \tl_if_in:NnTF \l__fnote_patch_tl { \hb@xt@ }
360     { \cs_set_eq:NN \__fnote_tmp:w \__fnote_patch_hb@xt@:w }
361     {
362       \tl_if_in:NnTF \l__fnote_patch_tl { \@makefnmark }
363       { \cs_set_eq:NN \__fnote_tmp:w \__fnote_patch_@makefnmark:w }
364       { \ERROR
365         \cs_set_eq:NN \__fnote_tmp:w \exp_stop_f: }
366     }
367   }
368   \tl_set:Nf \l__fnote_patch_tl
369   { \exp_after:wN \__fnote_tmp:w \l__fnote_patch_tl }
370   \cs_set:Npn \__fnote_tmp:w { \long \def \@makefntext ###1 }
371   \exp_after:wN \__fnote_tmp:w \exp_after:wN { \l__fnote_patch_tl }
372 }

```

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

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

```

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

```

Same for the other cases.

```

375 \cs_new:Npn \__fnote_patch_hb@xt@:w #1 \hb@xt@ #2 #
376 { \exp_stop_f: #1 \@makefntext@processX { \hb@xt@ #2 } }
377 \cs_new:Npn \__fnote_patch_@makefnmark:w #1 \@makefnmark
378 { \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:

```
379 \cs_new:Npn \@makefntext@processX #1#2{\UseSocket{tagsupport/fntext/mark}{#1{#2}}}
```

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

```

380
381 \AddToHook{begindocument}
382 {
383   \cs_if_eq:NNTF \@makefntext \old@std@class@makefntext
384   {
385     \cs_set_eq:NN \@makefntext \fnote_makefntext:n
386   }
387   {
388     \__fnote_patch:
389   }
390 }
391

```

```

392
393 % possibly add the following to check for multiple \hbox in
394 % the definition:
395 %
396 % \seq_set_split:NnV \l__fnote_patch_seq { \hbox } \l__fnote_patch_tl
397 % \int_compare:nT { \seq_count:N \l__fnote_patch_seq } > 2 \ERROR
398 %

```

5.13 Document-level commands

`\footnotetext`

```

399 \DeclareDocumentCommand\footnotetext {o+m}
400 {
401   \fnote_set_fnmark:nn {#1} \@mpfn
402   \@footnotetext {#2}
403 }

```

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

`\footnote`

```

404 \DeclareDocumentCommand\footnote {o+m}
405 {
406   \fnote_step_fnmark:nn {#1} \@mpfn
407   \cs_set_eq:NN \__fnote_footnotemark_finish: \prg_do_nothing:
408   \@footnotemark
409   \cs_set_eq:NN \__fnote_footnotemark_finish: \__fnote_footnotemark_default_finish:
410   \@footnotetext {#2}
411   \__fnote_footnotemark_finish:
412 }

```

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

`\footnotemark`

```

413 \DeclareDocumentCommand\footnotemark {o}
414 {
415   \fnote_step_fnmark:nn {#1} { footnote }
416   \@footnotemark
417 }

```

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

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

```

418 \DeclareDocumentCommand\footref {m}
419 {
420   \begingroup
421     \unrestored@protected@xdef\@thefnmark{\ref*{#1}}%
422   \endgroup
423   \bool_set_false:N \l__fnote_autodetect_bool
424   \tl_set:Nn \l__fnote_currentlabel_tl {#1}
425   \@footnotemark
426   \bool_set_true:N \l__fnote_autodetect_bool
427 }

```

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

5.14 Firstaid for packages and classes

5.15 Kernel patches

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

```

428 \NewSocketPlug{fntext/process}{mp}
429 {
430   \global\setbox\@mpfootins\vbox{%
431     \unvbox\@mpfootins
432     #1
433   }
434 }
```

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.

```

435 \AddToHook{class/memoir/before}
436 { \let\new@std@class@makecol\@makecol }
437 \AddToHook{class/memoir/after}
438 {
439   \cs_set_eq:NN \@footnotemark \fnote_footnotemark:
440   \cs_set_eq:NN \@makefntext\old@std@class@makefntext
441   \cs_set_eq:NN \@makecol\new@std@class@makecol
442 }
```

5.15.2 setspace

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

```

443 \AddToHook{package/setspace/after}
444 {\let \@footnotetext \fnote_footnotetext:n
445   \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.

```

446 \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.

```

447 \AtBeginDocument
448 {
449   \providecommand\hyper@linkstart{\@gobbletwo}
450   \providecommand\hyper@linkend{\@empty}
451 }
```


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).

```

452 \AtBeginDocument
453 {
454   \@ifpackageloaded{hyperref}
455   {
456     \legacy_if:nF{Hy@hyperfootnotes}{\bool_set_false:N \l_fnote_link_bool}
457   }
458   {
459     \bool_set_false:N \l_fnote_link_bool
460   }
461 }

```

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 inbuilt 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`

```

462 \ref_attribute_gset:nnnn {fnote/struct}{1}{now}{\l__fnote_currentstruct_tl}

```

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

```

\__fnote_label_hook:n

```

```

463 \cs_new_protected:Npn \__fnote_label_hook:n #1 {}
464 \AddToHookWithArguments{label}{\__fnote_label_hook:n{#1}}

```

(End of definition for `__fnote_label_hook:n`.)

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.

```

465 \AddToHook{fntext/begin}
466 {
467   \cs_set_protected:Npn \__fnote_label_hook:n #1
468   {
469     \ref_label:nn {#1} {fnote/struct}
470   }
471 }

```

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!

```

472 \cs_new_protected:Npn \fnote_class_new:nn #1 #2 % #1 name, #2 options
473 {
474   \prop_new:c { g__fnote_currentmarks_ #1 _prop }
475 }
476
477 \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).

```

478 \cs_new_protected:Npn \fnote_mark_gput:nn #1 #2 % #1 the representation of the mark, #2 type
479 {
480   \prop_gput:cxn { g__fnote_currentmarks_ #2 _prop }
481     { \tag_get:n{struct_num} }
482     { #1 }
483 }
484
485 \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.

```

486 \cs_new_protected:Npn \fnote_mark_gpop_all:nnN #1 #2 #3
487 {
488   \seq_clear:N #3
489   \prop_set_eq:Nc \l__fnote_tmpa_prop { g__fnote_currentmarks_ #2 _prop }
490   \prop_map_inline:Nn \l__fnote_tmpa_prop
491     {
492       \tl_if_eq:nnT {#1} { ##2 }
493       {

```

store the key (the structure number) in the seq

```

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

```

remove entry as used from the global prop

```

495         \prop_gremove:cn { g__fnote_currentmarks_ #2 _prop } {##1}
496       }
497     }
498   }
499   \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.

```
500 \NewSocketPlug{tagsupport/fnmark}{FEMark}
501 {
```

End an open mc and start the structure.

```
502 \tag_mc_end_push:
503 \tag_struct_begin:n { tag=footnotemark }
```

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

```
504 \bool_if:NTF \l__fnote_autodetect_bool
505 {
```

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.

```
506 \fnote_mark_gput:oo {\@thefnmark}{\l__fnote_type_tl}
507 \tl_set:Nx \l__fnote_linktarget_tl {footnote*.\tag_get:n{struct_num}}
508 }
```

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

```
509 {
510 \hook_gput_code:nne {tagpdf/finish/before} {tagpdf/footnote}
511 {
512 \exp_not:N\fnote_gput_refs:ee
513 { \tag_get:n{struct_num} }
514 { \ref_value:Nn\l__fnote_currentlabel_tl {fnote/struct} }
515 }
```

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

```
516 \tl_set:Nx \l__fnote_linktarget_tl {footnote*.\ref_value:Nn\l__fnote_currentlabel_tl}
517 }
```

And now the actual content

```
518 \tag_mc_begin:n{tag=Lbl}
519 %
520 \bool_if:NTF \l__fnote_link_bool
521 {
522 \exp_args:No
523 \hyper@linkstart
524 { \l__fnote_link_type_tl }
525 { \l__fnote_linktarget_tl }
526 #1
527 \hyper@linkend
528 }
529 { #1 }
530 \tag_mc_end:
531 \tag_struct_end:
532 \tag_mc_begin_pop:n{ }
533 }
```

At last assign the plug:

```
534 \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
535 \cs_new_protected:Npn \__fnote_gput_ref:nn #1 #2 %#1 the structure number receiving the ref
536 {
537   \tag_if_active:T
538   {
539     \tag_struct_gput:nne {#1}{ref}{\tag_struct_object_ref:e { #2 }}
540   }
541 }
542 \cs_new_protected:Npn \fnote_gput_refs:nn #1 #2 % pair of numbers
543 {
544   \__fnote_gput_ref:nn {#1}{#2}
545   \__fnote_gput_ref:nn {#2}{#1}
546 }
547 \cs_generate_variant:Nn \fnote_gput_refs:nn {ee}

```

(End of definition for __fnote_gput_ref:nn and \fnote_gput_refs:nn. This function is documented on page ??.)

```

548 \tl_new:N \l__fnote_dflt_struct_tl
549 \tl_set:Nn \l__fnote_dflt_struct_tl {1}

```

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

FENote (*plug*)

```

550 \NewSocketPlug{tagssupport/fntext/begin}{FENote}
551 {
552   \tag_mc_end_push:
test if a footnote is allowed, if not move up to the document structure.
553   \tag_check_child:nnTF {FENote}{pdf2}
554   {
555     \tag_struct_begin:n { tag=footnote }
556   }
557   {
558     \tag_struct_begin:n { tag=footnote,parent=\l__fnote_dflt_struct_tl }
559   }

```

Store the current structure number for labels.

```
560 \tl_set:Nx \l__fnote_currentstruct_tl { \tag_get:n{struct_num} }
```

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

```
561 \tag_struct_begin:n { tag=footnotelabel }\tag_struct_end:
```

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

```

562 \bool_if:NTF \l__fnote_autodetect_bool
563 {

```

find open marks with identical \@thefnmark

```
564 \fnote_mark_gpop_all:ooN { \@thefnmark }{ \l_fnote_type_t1 } \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:

```
565 \seq_map_inline:Nn \l__fnote_currentrefs_seq
566 {
567 \fnote_gput_refs:ee {##1}{ \l__fnote_currentstruct_t1 }
568 }
569 }
```

If no auto detection is done

```
570 {%no auto
571
572 }
573 }
```

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.

```
574 \NewSocketPlug{taggsupport/fntext/end}{FENote}
575 {
576 \tag_struct_end:
577 \tag_mc_begin_pop:n{ }
578 }
```

At last assign the plugs:

```
579 \AssignSocketPlug{taggsupport/fntext/begin}{FENote}
580 \AssignSocketPlug{taggsupport/fntext/end}{FENote}
```

The kernel socket taggsupport/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.

```
581 \NewSocketPlug{taggsupport/fntext/mark}{FENoteLbl}
582 {
583 \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).

```
584 %\seq_show:N\l__fnote_currentrefs_seq
585 \seq_map_inline:Nn\l__fnote_currentrefs_seq {\MakeLinkTarget*{footnote*.##1}}
586 \MakeLinkTarget*{footnote*.\l__fnote_currentstruct_t1}
```

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

```
587 \tag_struct_begin:n { tag=NonStruct,parent=\l__fnote_currentstruct_t1 +1 }
588 \tag_mc_begin:n { tag=Lbl }
589 #1
590 \tag_mc_end:
591 \tag_struct_end:
592 \tag_mc_begin_pop:n{ }
593 }
```

```
594 \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?

```
595 \NewSocketPlug{tagsupport/fntext/text}{FENotetext}
596 {
597   \tag_mc_end_push:
598   \tag_mc_begin:n{}
599   #1
600   \tag_mc_end:
601   \tag_mc_begin_pop:n{}
602 }
603 \AssignSocketPlug{tagsupport/fntext/text}{FENotetext}
```

```
604 \ExplSyntaxOff
605 </kernel>
606 <@@=>
```

6 Reimplementing the `footmisc` package

```
607 <*footmisc>
608 %%
609 %% Copyright (c) 1995-2011 Robin Fairbairns
610 %% Copyright (c) 2018-2023 Robin Fairbairns, Frank Mittelbach
611 %%
612 %% This file is part of the ‘latex-lab Bundle’.
613 %% -----
614 %%
615 %% It may be distributed and/or modified under the
616 %% conditions of the LaTeX Project Public License, either version 1.3c
617 %% of this license or (at your option) any later version.
618 %% The latest version of this license is in
619 %%   https://www.latex-project.org/lppl.txt
620 %% and version 1.3c or later is part of all distributions of LaTeX
621 %% version 2008 or later.
622 %%
623 %% This work has the LPPL maintenance status ‘maintained’.
624 %%
625 \NeedsTeXFormat{LaTeX2e}
626 \providecommand\DeclareRelease[3]{}
627 \providecommand\DeclareCurrentRelease[2]{}
628
629 \DeclareRelease{v5}{2011-06-06}{footmisc-2011-06-06.sty}
630 \DeclareCurrentRelease{}{2022-02-14}
631 \ProvidesPackage{latex-lab-footmisc}%
632   [2022/03/08 v6.0d
633     a miscellany of footnote facilities -- latex-lab version%
634   ]
635
636 \NeedsTeXFormat{LaTeX2e}[2020/10/01]
```

```

637 \newtoks\FN@temptoken
638 \providecommand\protected@writeaux{%
639   \protected@write\@auxout
640 }
641 \def\l@advance@macro{\@advance@macro\edef}
642 \def\@advance@macro#1#2#3{\expandafter\@tempcnta#2\relax
643   \advance\@tempcnta#3\relax
644   #1#2{\the\@tempcnta}%
645 }
646 \let\@advance@macro\l@advance@macro
647 \DeclareOption{symbol}{\renewcommand\thefootnote{\fnsymbol{footnote}}}
648 \newif\ifFN@robust \FN@robustfalse
649 \DeclareOption{symbol*}{%
650   \renewcommand\thefootnote{\@fnsymbol\c@footnote}%
651   \FN@robusttrue
652   \AtEndOfPackage{\setfnsymbol{lamport*-\robust}}}%
653 }
654 \newif\ifFN@para \FN@parafalse
655 \DeclareOption{para}{%

```

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

```

656 %   \PackageError{footmisc}{Option "\CurrentOption" incompatible with
657 %                                     option "side"}%
658 %                                     {I shall ignore "\CurrentOption"}%
659 \FN@paratrue
660 \setlength\footnotemargin{-\maxdimen} % default when para is used
661 }
662 \DeclareOption{side}{\ifFN@para
663   \PackageError{footmisc}{Option "\CurrentOption" incompatible with
664   option "para"}%
665   {I shall ignore "\CurrentOption"}%
666 \else
667   \AddToHook{fntext/para}{%
668     \hsize\marginparwidth % correct the default \hsize
669     \footnotesep\z@ % don't add a default separation
670   }
671   \AssignSocketPlug{fntext/process}{side}
672 % \AssignSocketPlug{fntext/make}{default}
673 \AssignSocketPlug{fntext/begin}{noop}
674 \AssignSocketPlug{fntext/end}{noop}
675 \fi
676 }
677 \let\footnotelayout\@empty
678 \DeclareOption{ragged}{%
679   \@ifundefined{RaggedRight}%
680   {\renewcommand\footnotelayout{\linepenalty50 \raggedright}}%
681   {\renewcommand\footnotelayout{\linepenalty50 \RaggedRight}}%
682 }
683 \newif\ifFN@perpage
684 \FN@perpagefalse
685 \DeclareOption{perpage}{%
686   \FN@perpagetrue
687 }

```

```

688 \newif\ifFN@fixskip \FN@fixskipfalse
689
690 \let\FN@bottomcases\thr@@
691 \newif\ifFN@abovefloats \FN@abovefloatstrue
692 \DeclareOption{bottom}{%
693 \let\FN@bottomcases\@ne
694 \FN@abovefloatsfalse
695 \FN@fixskiptrue
696 }
697 \DeclareOption{bottomfloats}{%
698 \let\FN@bottomcases\tw@
699 \FN@abovefloatstrue \FN@fixskiptrue
700 }
701 \DeclareOption{abovefloats}{\FN@abovefloatstrue \FN@fixskiptrue}
702 \DeclareOption{belowfloats}{\FN@abovefloatsfalse \FN@fixskiptrue}
703 \DeclareOption{marginal}{%
704 \footnotemargin-0.8em\relax
705 }
706 \DeclareOption{flushmargin}{%
707 \footnotemargin0pt\relax
708 }
709 \newif\ifFN@hangfoot \FN@hangfootfalse
710 \DeclareOption{hang}{%
711 \FN@hangfoottrue
712 }
713 \newcommand*\hangfootparskip{0.5\baselineskip}
714 \newcommand*\hangfootparindent{0em}%
715 \DeclareOption{norule}{%
716 \renewcommand\footnoterule{}%
717 \advance\skip\footins 4\p@\@plus2\p@\relax
718 }
719 \DeclareOption{splitrule}{%
720 \gdef\split@prev{0}
721 \let\pagefootnoterule\footnoterule
722 \let\mpfootnoterule\footnoterule
723 \def\splitfootnoterule{\kern-3\p@ \hrule \kern2.6\p@}
724 \def\footnoterule{\relax
725 \ifx \@listdepth\@mplistdepth
726 \mpfootnoterule
727 \else
728 \ifnum\split@prev=\z@
729 \pagefootnoterule
730 \else
731 \splitfootnoterule
732 \fi
733 \xdef\split@prev{\the\insertpenalties}%
734 \fi
735 }%
736 }
737 \newif\ifFN@stablefootnote \FN@stablefootnotefalse
738 \DeclareOption{stable}{\FN@stablefootnotetrue}
739 \newif\ifFN@multiplefootnote \FN@multiplefootnotefalse
740 \DeclareOption{multiple}{\FN@multiplefootnotetrue}
741 \ProcessOptions

```


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

```

742 \ifFN@para
743   \NewSocketPlug{@makecol/footnotes}{para}{%
744     \global\setbox\footins\vbox{\FN@makefootnoteparagraph}%
745   }
746   \AssignSocketPlug{@makecol/footnotes}{para}
747 \fi
748 \ifFN@fixskip
749   \def\@outputbox@removebskip{%
750     \ifx\@textbottom\relax \else
751       \@outputbox@append{%
752         \@tempskipa\lastskip
753         \ifnum \gluestretchorder\@tempskipa>\z@
754           \vskip-\@tempskipa
755           \xdef\@outputbox@reinsertbskip
756             {\noexpand\@outputbox@append{\vskip\the\@tempskipa}}%
757         \else
758           \global\let\@outputbox@reinsertbskip\relax
759         \fi
760       }%
761     \fi
762   }
763   \let\@outputbox@reinsertbskip\relax
764 \else
765   \let\@outputbox@removebskip \relax
766   \let\@outputbox@reinsertbskip\relax
767 \fi
768 \ifcase \FN@bottomcases\relax
769   \ERROR
770 \or %1
771   \ifFN@abovefloats
772     \AssignSocketPlug {@makecol/outputbox}{space-footnotes-floats}
773   \else
774     \AssignSocketPlug {@makecol/outputbox}{floats-footnotes-space}
775   \fi
776 \or %2
777   \ifFN@abovefloats
778     \AssignSocketPlug {@makecol/outputbox}{footnotes-space-floats}
779   \else
780     \AssignSocketPlug {@makecol/outputbox}{space-floats-footnotes}
781   \fi
782 \or %3
783   \ifFN@abovefloats
784     \AssignSocketPlug {@makecol/outputbox}{footnotes-floats}
785   \else
786     \AssignSocketPlug {@makecol/outputbox}{floats-footnotes}
787   \fi
788 \else
789   \ERROR
790 \fi
791
792 % next can be dropped when cleaned up

```

```

793 \newif\ifFN@setspace
794 \@ifpackageloaded{setspace}%
795 {%
796   \FN@setspacetrue
797   \@ifclassloaded{memoir}%
798   {%
799     \AddToHook{fntext}{\let\baselinestretch\m@m@singlespace}%
800     \let\FN@baselinestretch\m@m@singlespace
801   }%
802   {%
803     \AddToHook{fntext}{\let\baselinestretch\setspace@singlespace}%
804     \let\FN@baselinestretch\setspace@singlespace
805   }%
806 }%
807 {%
808   \FN@setspacefalse
809 }
810
811
812
813 \ifFN@para
814
815 % \AssignSocketPlug{fntext/process}{default}
816 \AssignSocketPlug{fntext/make}{para}
817 \AssignSocketPlug{fntext/begin}{noop}
818 \AssignSocketPlug{fntext/end}{para}
819
820 \fi
821
822
823
824 \ifFN@para
825   \let\FN@tempboxa\@tempboxa
826   \newbox\FN@tempboxb
827   \newbox\FN@tempboxc
828   \newskip\footglue \footglue=1em plus.3em minus.3em
829
830   %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
831   \newdimen\footnotebaselineskip
832
833   % establish late:
834
835   \AddToHook{begindocument/before} {%
836     {%
837       \footnotesize
838       \global\footnotebaselineskip=\normalbaselineskip
839     }%
840   }

```

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.

```

841
842 \long\def\FN@makefootnoteparagraph{%
843   \FN@setfootnoteparawidth
844   \@parboxrestore

```

```

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

```

Fix a bug in perpage ...

```

870   \def\@stpelt#1{\global\csname c@#1\endcsname \m@ne
871     \stepcounter{#1}%
872     \pp@fix@MakePerPage{#1}%
873   }
874   \def\pp@fix@MakePerPage#1{%
875     \ifnum \value{#1}>\z@
876       \addtocounter{#1}\m@ne\fi
877   }

```

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`:

```

878   \def\pp@c1@end@iii\stepcounter#1\pp@fix@MakePerPage#2{}
879   \fi
880
881
882 \ifFN@para
883
884 % This can use the default interface, except that a negative value for
885 % \footnotemargin makes little sense, so we test for this and warn if
886 % necessary. But -\maxdimen is ok again, so would need to be a litte bit more elaborate.
887 %
888
889 %\AddToHook{fntext/para}{

```

```

890 % \ifdim \footnotemargin >\z@ \else
891 %   \PackageWarningNoline{footmisc}{Option 'para' needs positive \noexpand\footnotemargin}%
892 %   \footnotemargin 1.8em\relax
893 % \fi
894 %}
895
896
897 \AddToHook{fntext/begin}{\nobreak \hspace{.2em}}
898
899 \else
900
901 \ifFN@hangfoot
902   \long\def\@makefntext#1{%
903     \bgroup
904       \setbox\@tempboxa\hbox{%
905         \ifdim\footnotemargin>\z@
906           \hb@xt@\footnotemargin{\@makefnmark\hss}%
907         \else
908           \@makefnmark
909         \fi
910       }%
911       \leftmargin\wd\@tempboxa
912       \rightmargin\z@
913       \linewidth \columnwidth
914       \advance \linewidth -\leftmargin
915       \parshape \@ne \leftmargin \linewidth
916       \footnotesize
917       \@setpar{\@par}}%
918       \leavevmode
919       \llap{\box\@tempboxa}%
920       \parskip\hangfootparskip\relax
921       \parindent\hangfootparindent\relax
922       \footnotelayout#1%
923     \par
924   \egroup
925 }
926
927 \else
928
929 % This is now using the default interface:
930 %
931 % \long\def\@makefntext#1{%
932 %   \parindent1em
933 %   \noindent
934 %   \ifdim\footnotemargin>\z@
935 %     \hb@xt@\footnotemargin{\hss\@makefnmark}%
936 %   \else
937 %     \ifdim\footnotemargin=\z@
938 %       \llap{\@makefnmark}%
939 %     \else
940 %       \llap{\hb@xt@ -\footnotemargin{\@makefnmark\hss}}%
941 %     \fi
942 %   \fi
943 %   \footnotelayout#1%

```

```

944 % }
945
946 \fi
947 \fi
948
949
950
951
952 \ifFN@multiplefootnote
953 \providecommand*{\multiplefootnotemark}{3sp}
    We tag the separator as artifact
    TODO: why is this done with \providecommand?

954 \ExplSyntaxOn
955 \providecommand*{\multfootsep}{\tag_mc_end_push:\tag_mc_begin:n{artifact},\tag_mc_end:\tag
956 \ExplSyntaxOff
957 \AddToHook{fnmark}      {\FN@mf@check}
958 \AddToHook{fnmark/end}  {\FN@mf@prepare}
959 %
960 \def\FN@mf@prepare{%
961     \kern-\multiplefootnotemark
962     \kern\multiplefootnotemark\relax
963 }
964 \def\FN@mf@check{%
965     \ifdim\lastkern=\multiplefootnotemark\relax
966 %?? is that necessary or even correct ??
967     \edef\x@sf{\the\spacefactor}%
968 %?? shouldn't that be 2 unkerns ?? (none would also be ok)
969     \unkern % new
970     \unkern
971     \textsuperscript{\multfootsep}%
972     \spacefactor\x@sf\relax
973 \fi
974 }
975 \else
976 \let\FN@mf@prepare\relax
977 \fi
978 \ifFN@stablefootnote
979 \let\FN@sffootnote\footnote
980 \def\footnote{\ifx\protect\@typeset@protect
981     \expandafter\FN@sffootnote
982 \else
983     \expandafter\FN@sfgobble@opt
984 \fi
985 }
986 \edef\FN@sfgobble@opt{\noexpand\protect
987     \expandafter\noexpand\csname FN@sfgobble@opt \endcsname}
988 \expandafter\def\csname FN@sfgobble@opt \endcsname{%
989     \@ifnextchar[%]
990         \FN@sfgobble@twobracket
991         \@gobble
992 }
993 \def\FN@sfgobble@twobracket[#1]#2{}
994 \let\FN@sffootnotemark\footnotemark

```

```

995 \def\footnotemark{\ifx\protect\@typeset@protect
996   \expandafter\FN@sffootnotemark
997   \else
998     \expandafter\FN@sfgobble@optonly
999   \fi
1000 }
1001 \edef\FN@sfgobble@optonly{\noexpand\protect
1002   \expandafter\noexpand\csname FN@sfgobble@optonly \endcsname}
1003 \expandafter\def\csname FN@sfgobble@optonly \endcsname{%
1004   \ifnextchar[%]
1005     \FN@sfgobble@bracket
1006     {%}%
1007   }
1008 \def\FN@sfgobble@bracket[#1]{%
1009   \fi
1010   \newcommand\setfnsymbol[1]{%
1011     \@bsphack
1012     \@ifundefined{FN@fnsymbol@#1}{%
1013       {%
1014         \PackageError{footmisc}{Symbol style "#1" not known}%
1015         \@eha
1016       }{%
1017         \expandafter\let\expandafter@\fnsymbol\csname
1018           FN@fnsymbol@#1\endcsname
1019       }%
1020       \@esphack
1021     }
1022     \let\FN@fnsymbol@lport@\fnsymbol
1023     \newif\if@tempswb
1024     \DeclareDocumentCommand\DefineFNSymbols {sm0{text}m}{%
1025       \expandafter\ifx\csname FN@fnsymbol@#2\endcsname\relax
1026         \PackageInfo{footmisc}{Declaring symbol style #2}%
1027       \else
1028         \PackageWarning{footmisc}{Redeclaring symbol style #2}%
1029       \fi
1030       \toks@{}%
1031       \def\@tempb{\end}%
1032       \FN@build@symboldef#4\end
1033       \def\@tempc{math}%
1034       \def\@tempd{#3}%
1035       \expandafter\xdef\csname FN@fnsymbol@#2\endcsname##1{%
1036         \ifx\@tempc\@tempd
1037           \noexpand\ensuremath
1038         \else
1039           \noexpand\nfss@text
1040         \fi
1041         {%
1042           \noexpand\ifcase##1%
1043             \the\toks@
1044           \noexpand\else
1045             \IfBooleanTF#1{\noexpand\@ctrerr}%
1046             {\noexpand\FN@orange##1}%
1047           \noexpand\fi
1048         }%

```

```

1049 }%
1050 }
1051 \def\FN@build@symboldef#1{%
1052   \def\@tempa{#1}%
1053   \ifx\@tempa\@tempb
1054     \else
1055       \toks@\expandafter{\the\toks@\or#1}%
1056       \expandafter\FN@build@symboldef
1057     \fi
1058 }
1059 \DeclareDocumentCommand\DefineFNsymbolsTM {smm}{%
1060   \expandafter\ifx\csname FN@fnsymbol@#2\endcsname\relax
1061     \PackageInfo{footmisc}{Declaring symbol style #2}%
1062   \else
1063     \PackageWarning{footmisc}{Redeclaring symbol style #2}%
1064   \fi
1065   \toks@{}%
1066   \def\@tempb{\end}%
1067   \FN@build@symboldefTM#3\end\@null
1068   \expandafter\xdef\csname FN@fnsymbol@#2\endcsname##1{%
1069     \noexpand\ifcase##1%
1070       \the\toks@
1071     \noexpand\else
1072       \IfBooleanTF#1{\noexpand\@ctrerr}%
1073       {\noexpand\FN@orange##1}%
1074     \noexpand\fi
1075   }%
1076 }
1077 \def\FN@build@symboldefTM#1#2{%
1078   \def\@tempa{#1}%
1079   \ifx\@tempa\@tempb
1080     \else
1081       \toks@\expandafter{\the\toks@\or\TextOrMath{#1}{#2}}%
1082       \expandafter\FN@build@symboldefTM
1083     \fi
1084 }
1085 \def\FN@orange#1{%
1086   \ifFN@robust
1087     \@arabic#1%
1088     \@bsphack
1089     \PackageInfo{footmisc}{Footnote number \number#1 out of range}%
1090     \protect\@fnsymbol@orange
1091     \@esphack
1092   \else \@ctrerr \fi
1093 }
1094 \global\let\@diagnose@fnsymbol@orange\relax
1095 \AtEndDocument{\@diagnose@fnsymbol@orange}
1096 \def\@fnsymbol@orange{%
1097   \gdef\@diagnose@fnsymbol@orange{%
1098     \PackageWarningNoLine{footmisc}{Some footnote number(s)
1099       were out of range
1100       \MessageBreak
1101       see log for details%
1102   }%

```

```

1103 }%
1104 }
1105 \DefineFNSymbolsTM{bringhurst}{%
1106   \textasteriskcentered *%
1107   \textdagger \dagger
1108   \textdaggerdbl \ddagger
1109   \textsection \mathsection
1110   \textbardbl \||%
1111   \textparagraph \mathparagraph
1112 }%
1113 \DefineFNSymbolsTM{chicago}{%
1114   \textasteriskcentered *%
1115   \textdagger \dagger
1116   \textdaggerdbl \ddagger
1117   \textsection \mathsection
1118   \textbardbl \||%
1119   \#\#%
1120 }%
1121 \DefineFNSymbolsTM{wiley}{%
1122   \textasteriskcentered *%
1123   {\textasteriskcentered\textasteriskcentered}{**}%
1124   \textdagger \dagger
1125   \textdaggerdbl \ddagger
1126   \textsection \mathsection
1127   \textparagraph \mathparagraph
1128   \textbardbl \||%
1129 }%
1130 \DefineFNSymbolsTM{lamport-robust}{%
1131   \textasteriskcentered *%
1132   \textdagger \dagger
1133   \textdaggerdbl \ddagger
1134   \textsection \mathsection
1135   \textparagraph \mathparagraph
1136   \textbardbl \||%
1137   {\textasteriskcentered\textasteriskcentered}{**}%
1138   {\textdagger\textdagger}{\dagger\dagger}%
1139   {\textdaggerdbl\textdaggerdbl}{\ddagger\ddagger}%
1140 }
1141 \DefineFNSymbolsTM*{lamport*}{%
1142   \textasteriskcentered *%
1143   \textdagger \dagger
1144   \textdaggerdbl \ddagger
1145   \textsection \mathsection
1146   \textparagraph \mathparagraph
1147   \textbardbl \||%
1148   {\textasteriskcentered\textasteriskcentered}{**}%
1149   {\textdagger\textdagger}{\dagger\dagger}%
1150   {\textdaggerdbl\textdaggerdbl}{\ddagger\ddagger}%
1151   {\textsection\textsection}{\mathsection\mathsection}%
1152   {\textparagraph\textparagraph}{\mathparagraph\mathparagraph}%
1153   {\textasteriskcentered\textasteriskcentered\textasteriskcentered}{***}%
1154   {\textdagger\textdagger\textdagger}{\dagger\dagger\dagger}%
1155   {\textdaggerdbl\textdaggerdbl\textdaggerdbl}{\ddagger\ddagger\ddagger}%
1156   {\textsection\textsection\textsection}%%

```



```

1157     {\mathsection\mathsection\mathsection}%
1158     {\textparagraph\textparagraph\textparagraph}%%
1159     {\mathparagraph\mathparagraph\mathparagraph}%
1160 }
1161 \setfnsymbol{lampport*}
1162 \DefineFNsymbolsTM{lampport*--robust}{%
1163     \textasteriskcentered *%
1164     \textdagger      \dagger
1165     \textdaggerdbl   \ddagger
1166     \textsection     \mathsection
1167     \textparagraph   \mathparagraph
1168     \textbardbl      \|%
1169     {\textasteriskcentered\textasteriskcentered}{**}%
1170     {\textdagger\textdagger}{\dagger\dagger}%
1171     {\textdaggerdbl\textdaggerdbl}{\ddagger\ddagger}%
1172     {\textsection\textsection}{\mathsection\mathsection}%
1173     {\textparagraph\textparagraph}{\mathparagraph\mathparagraph}%
1174     {\textasteriskcentered\textasteriskcentered\textasteriskcentered}{***}%
1175     {\textdagger\textdagger\textdagger}{\dagger\dagger\dagger}%
1176     {\textdaggerdbl\textdaggerdbl\textdaggerdbl}{\ddagger\ddagger\ddagger}%
1177     {\textsection\textsection\textsection}%%
1178     {\mathsection\mathsection\mathsection}%
1179     {\textparagraph\textparagraph\textparagraph}%%
1180     {\mathparagraph\mathparagraph\mathparagraph}%
1181 }
1182 \newcommand\mpfootnotemark{%
1183     \@ifnextchar[%]
1184     \@xmpfootnotemark
1185     {%
1186         \stepcounter\@mpfn
1187         \protected@xdef\@thefnmark{\thempfn}%
1188         \@footnotemark
1189     }%
1190 }
1191 \def\@xmpfootnotemark[#1]{%
1192     \begingroup
1193         \csname c@\@mpfn\endcsname #1\relax
1194         \unrestored@protected@xdef\@thefnmark{\thempfn}%
1195     \endgroup
1196     \@footnotemark
1197 }

```

TEMP PATCHES FOR TESTING

```

1198 \endinput
1199 </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	
\#	1119
\	1110, 1118, 1128, 1136, 1147, 1168
A	
\addtocounter	876
\AddToHook	308, 381, 435, 437, 443, 445, 465, 667, 799, 803, 835, 889, 897, 957, 958
\AddToHookWithArguments	464
\advance	643, 717, 914
\AssignSocketPlug	281, 293, 296, 305, 342, 534, 579, 580, 594, 603, 671, 672, 673, 674, 746, 772, 774, 778, 780, 784, 786, 815, 816, 817, 818
\AtBeginDocument	315, 447, 452
\AtEndDocument	1095
\AtEndOfPackage	652
B	
\baselineskip	713, 845
\baselinestretch	445, 799, 803
\begingroup	420, 1192
\bgroup	903
bool commands:	
\bool_gset_true:N	119
\bool_if:NTF	122, 134, 504, 520, 562
\bool_new:N	69, 76, 118
\bool_set_false:N	423, 456, 459
\bool_set_true:N	70, 77, 426
\box	291, 919
C	
\columnwidth	16, 228, 290, 864, 865, 913
\cs	236, 237
cs commands:	
\cs_generate_variant:Nn	80, 81, 82, 83, 485, 499, 547
\cs_gset_eq:NN	129, 153
\cs_if_eq:NNTF	383
\cs_new:Npn	373, 375, 377, 379
\cs_new_eq:NN	198, 351
\cs_new_protected:Npn	84, 97, 120, 132, 156, 183, 201, 321, 353, 463, 472, 478, 486, 535, 542
\cs_set:Npn	370
\cs_set_eq:NN	200, 310, 357, 360, 363, 365, 385, 407, 409, 439, 440, 441
\cs_set_protected:Npn	467
\csname	20, 870, 987, 988, 1002, 1003, 1017, 1025, 1035, 1060, 1068, 1193
\CurrentOption	656, 658, 663, 665
D	
\dagger	1107, 1115, 1124, 1132, 1138, 1143, 1149, 1154, 1164, 1170, 1175
\ddagger	1108, 1116, 1125, 1133, 1139, 1144, 1150, 1155, 1165, 1171, 1176
\DeclareCurrentRelease	627, 630
\DeclareDocumentCommand	399, 404, 413, 418, 1024, 1059
\DeclareOption	647, 649, 655, 662, 678, 685, 692, 697, 701, 702, 703, 706, 710, 715, 719, 738, 740
\DeclareRelease	626, 629
\def	12, 18, 29, 30, 34, 231, 370, 446, 641, 642, 723, 724, 749, 842, 849, 861, 865, 870, 874, 878, 902, 931, 960, 964, 980, 988, 993, 995, 1003, 1008, 1031, 1033, 1034, 1051, 1052, 1066, 1077, 1078, 1085, 1096, 1191
default (plug)	5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 6, 20
\DefineFNsymbols	1024
\DefineFNsymbolsTM	1059, 1105, 1113, 1121, 1130, 1141, 1162
\dimexpr	289
\dp	226, 287
E	
\edef	165, 641, 967, 986, 1001
\egroup	924
\else	330, 333, 336, 666, 727, 730, 750, 757, 764, 773, 779, 785, 788, 852, 864, 890, 899, 907, 927, 936, 939, 975, 982, 997, 1027, 1038, 1044, 1054, 1062, 1071, 1080, 1092
\end	1031, 1032, 1066, 1067
\endcsname	20, 870, 987, 988, 1002, 1003, 1018, 1025, 1035, 1060, 1068, 1193
\endgroup	422, 1195
\endinput	1199
\ensurermath	1037
\ERROR	21, 351, 364, 397, 769, 789
exp commands:	
\exp_after:wN	369, 371
\exp_args:No	522
\exp_not:N	512
\exp_stop_f:	365, 374, 376, 378

\expandafter	642, 981, 983, 987, 988, 996, 998, 1002, 1003, 1017, 1025, 1035, 1055, 1056, 1060, 1068, 1081, 1082
\ExplSyntaxOff	604, 956
\ExplSyntaxOn	64, 954
F	
FEMark (plug)	500
FENote (plug)	550
FENoteLbl (plug)	581
FENotetext (plug)	595
\fi	174, 192, 319, 338, 339, 340, 675, 732, 734, 747, 759, 761, 767, 775, 781, 787, 790, 820, 855, 857, 864, 876, 879, 893, 909, 941, 942, 946, 947, 973, 977, 984, 999, 1009, 1029, 1040, 1047, 1057, 1064, 1074, 1083, 1092
\floatingpenalty	227
fnmark (hook)	4, 4, 109
fnmark/after (hook)	4, 4, 4, 6, 16, 109
fnmark/before (hook)	4, 4, 4, 109
fnmark/begin (hook)	4, 4, 109
fnmark/end (hook)	4, 4, 16, 109
fnote commands:	
\fnote_class_new:nn	9, 472, 472, 477
\g_fnote_debug_bool	118, 119, 122, 134
\fnote_footnotemark:	156, 156, 200, 439
\fnote_footnotetext:n	201, 201, 310, 444
\fnote_gput_refs:nn	512, 535, 542, 547, 567
\l_fnote_link_bool	25, 69, 456, 459, 520
\l_fnote_link_type_tl	71, 524
\fnote_makefntext:n	10, 21, 22, 321, 321, 385
\fnote_mark_gpop_all:nnN	9, 486, 486, 499, 564
\fnote_mark_gput:nn	9, 478, 478, 485, 506
\fnote_set_fnmark:nn	97, 97, 401
\fnote_step_fnmark:nn	84, 84, 406, 415
\l_fnote_type_tl	67, 506, 564
fnote internal commands:	
\l__fnote_autodetect_bool	76, 423, 426, 504, 562
\l__fnote_currentlabel_tl	23, 27, 74, 424, 514, 516
\l__fnote_currentrefs_seq	75, 564, 565, 584, 585
\l__fnote_currentstruct_tl	25, 78, 79, 462, 560, 567, 586, 587
__fnote_debug_footnotemark:	120, 129, 157
__fnote_debug_footnotetext:	132, 153, 202
\l__fnote_dflt_struct_tl	548, 549, 558
__fnote_footmark_finish:	16
__fnote_footnotemark_default_- finish:	16, 183, 183, 198, 409
__fnote_footnotemark_finish:	16, 181, 183, 198, 407, 409, 411
__fnote_gput_ref:nn	535, 535, 544, 545
__fnote_label_hook:n	463, 463, 464, 467
\l__fnote_linktarget_tl	73, 507, 516, 525
__fnote_patch:	353, 388
__fnote_patch_@makefnmark:w	363, 377
__fnote_patch_hb@xt:w	360, 375
__fnote_patch_hbox:w	357, 373
\l__fnote_patch_seq	396, 397
\l__fnote_patch_tl	350, 355, 356, 359, 362, 368, 369, 371, 396
__fnote_tmp:w	351, 357, 360, 363, 365, 369, 370, 371
\l__fnote_tmpa_prop	65, 489, 490
\l__fnote_tmpa_tl	66
\fnsymbol	647
fntext (hook)	6, 6, 113
fntext/after (hook)	6, 6, 6, 113
fntext/before (hook)	6, 6, 6, 113
fntext/begin (hook)	6, 6, 6, 113
fntext/begin (socket)	5, 5, 294
fntext/end (hook)	6, 6, 6, 113
fntext/end (socket)	5, 5, 297
fntext/make (socket)	5, 5, 5, 5, 6, 282
fntext/mark (socket)	5, 5, 326
fntext/para (hook)	6, 6, 6, 6, 20, 113
fntext/process (socket)	5, 5, 278
fntext/text (socket)	6, 6, 343
\footglue	303, 828
\footins	5, 279, 717, 744, 846
\footnote	5, 8, 10, 19, 404, 979, 980
\footnotebaselineskip	290, 831, 838, 845
\footnotelayout	677, 680, 681, 922, 943
\footnotemargin	5, 7, 20, 312, 313, 317, 318, 328, 329, 331, 334, 337, 660, 704, 707, 885, 890, 891, 892, 905, 906, 934, 935, 937, 940
\footnotemark	4, 8-10, 413, 994, 995
\footnoterule	716, 721, 722, 724
\footnotesep	23, 225, 295, 669, 854
\footnotesize	15, 218, 837, 916
\footnotetext	5, 6, 8-10, 19, 399
\footref	9, 10, 23, 25, 29, 418

G		<code>\int_set:Nn</code> 92, 104
<code>\gdef</code> 720, 1097		<code>\interfootnotelinepenalty</code> 224
<code>\global</code> .. 13, 430, 744, 758, 838, 870, 1094		<code>\interlinepenalty</code> 6, 224
<code>\gluestretchorder</code> 753		
group commands:		K
<code>\group_begin:</code> 91, 103		<code>\kern</code> 723, 961, 962
<code>\group_end:</code> 94, 106		
H		L
<code>\hangfootparindent</code> 714, 921		<code>\label</code> 9
<code>\hangfootparskip</code> 713, 920		<code>\lastbox</code> 849
<code>\hbox</code> 21,		<code>\lastkern</code> 965
22, 34, 286, 356, 373, 374, 393, 396, 904		<code>\lastskip</code> 752
hook commands:		<code>\leavevmode</code> 163, 918
<code>\hook_gput_code:nnn</code> 82, 510		<code>\leftmargin</code> 911, 914, 915
<code>\hook_log:n</code> 124, 125, 126,		legacy commands:
127, 128, 147, 148, 149, 150, 151, 152		<code>\legacy_if:nTF</code> 456
<code>\hook_use:n</code> 161,		<code>\let</code> 436, 444, 445, 646,
171, 177, 205, 215, 240, 257, 264, 275		677, 690, 693, 698, 721, 722, 758,
Hooks:		763, 765, 766, 799, 800, 803, 804,
<code>fnmark</code> 4, 4, 109		825, 976, 979, 994, 1017, 1022, 1094
<code>fnmark/after</code> 4, 4, 4, 6, 16, 109		<code>\linepenalty</code> 680, 681
<code>fnmark/before</code> 4, 4, 4, 109		<code>\linewidth</code> 913, 914, 915
<code>fnmark/begin</code> 4, 4, 109		<code>\llap</code> 5, 7, 332, 337, 919, 938, 940
<code>fnmark/end</code> 4, 4, 16, 109		<code>\long</code> 12, 370, 842, 902, 931
<code>fnmark/after</code> 6, 6, 6, 113		<code>\ltlabfootnotedate</code> 4
<code>fnmark/before</code> 6, 6, 6, 113		<code>\ltlabfootnoteversion</code> 4
<code>fnmark/begin</code> 6, 6, 6, 113		
<code>fnmark/end</code> 6, 6, 6, 113		M
<code>fnmark/para</code> 6, 6, 6, 20, 113		<code>\MakeLinkTarget</code> 585, 586
<code>\hrule</code> 723		<code>\MakePerPage</code> 35, 869
<code>\hspace</code> 6, 18, 16, 228, 668, 862, 865		<code>\marginpar</code> 280
<code>\hskip</code> 303		<code>\marginparwidth</code> 668
<code>\hss</code> 329, 337, 349, 906, 935, 940		<code>\mathparagraph</code> 1111, 1127, 1135,
<code>\ht</code> 288		1146, 1152, 1159, 1167, 1173, 1180
I		<code>\mathsection</code> .. 1109, 1117, 1126, 1134,
<code>identity (plug)</code> 6, 16, 17, 21		1145, 1151, 1157, 1166, 1172, 1178
<code>\IfBooleanTF</code> 1045, 1072		<code>\maxdimen</code> ... 5, 7, 313, 317, 334, 660, 886
<code>\ifcase</code> 768, 1042, 1069		<code>\MessageBreak</code> 1100
<code>\ifdim</code> 317,		<code>mp (plug)</code> 5
328, 331, 334, 890, 905, 934, 937, 965		<code>\mpfootnotemark</code> 1182
<code>\ifhbox</code> 850		<code>\mpfootnoterule</code> 722, 726
<code>\ifhmode</code> 164, 190		<code>\multfootsep</code> 955, 971
<code>\ifnum</code> 728, 753, 862, 875		<code>\multiplefootnotemark</code> 953, 961, 962, 965
<code>\ifx</code> 725, 750,		
980, 995, 1025, 1036, 1053, 1060, 1079		N
<code>\ignorespaces</code> 23, 259		<code>\NeedsTeXFormat</code> 625, 636
<code>\insert</code> 5, 16, 279		<code>\newbox</code> 826, 827
<code>\insertpenalties</code> 733		<code>\newcommand</code> 346, 713, 714, 1010, 1182
int commands:		<code>\newdimen</code> 312, 831
<code>\int_compare:nTF</code> 397		<code>\NewHook</code> . 110, 111, 112, 114, 115, 116, 117
		<code>\newif</code> 648, 654,
		683, 688, 691, 709, 737, 739, 793, 1023
		<code>\NewMirroredHookPair</code> 109, 113
		<code>\newskip</code> 828

<code>\NewSocket</code>	199, 278, 282, 294, 297, 306, 307, 326, 343, 344, 345
<code>\NewSocketPlug</code> 279, 280, 283, 284, 295, 298, 299, 327, 428, 500, 550, 574, 581, 595, 743
<code>\newtoks</code>	637
<code>\nobreak</code>	4, 173, 897
<code>\noexpand</code>	756, 891, 986, 987, 1001, 1002, 1037, 1039, 1042, 1044, 1045, 1046, 1047, 1069, 1071, 1072, 1073, 1074
<code>\noindent</code>	322, 348, 933
<code>noop (plug)</code>	5, 5, 5, 6
<code>\normalbaselineskip</code>	838
<code>\normalfont</code>	34
<code>\number</code>	1089
O	
<code>\or</code>	770, 776, 782, 1055, 1081
P	
<code>\PackageError</code>	656, 663, 1014
<code>\PackageInfo</code>	1026, 1061, 1089
<code>\PackageWarning</code>	1028, 1063
<code>\PackageWarningNoLine</code>	1098
<code>\PackageWarningNoline</code>	891
<code>\pagefootnoterule</code>	721, 729
<code>\par</code>	24, 268, 923
<code>para (plug)</code>	5, 5, 5
<code>\parindent</code>	6, 230, 347, 921, 932
<code>\parshape</code>	915
<code>\parskip</code>	920
<code>\penalty</code>	302
Plugs:	
<code>default</code>	5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 6, 20
<code>FEMark</code>	500
<code>FENote</code>	550
<code>FENoteLbl</code>	581
<code>FENotetext</code>	595
<code>identity</code>	6, 16, 17, 21
<code>mp</code>	5
<code>noop</code>	5, 5, 5, 6
<code>para</code>	5, 5, 5
<code>side</code>	5
prg commands:	
<code>\prg_do_nothing:</code>	129, 153, 407
<code>\ProcessOptions</code>	741
prop commands:	
<code>\prop_gput:Nnn</code>	81, 480
<code>\prop_gremove:Nn</code>	495
<code>\prop_map_inline:Nn</code>	490
<code>\prop_new:N</code>	65, 474
<code>\prop_set_eq:NN</code>	489
<code>\protect</code>	980, 986, 995, 1001, 1090
<code>\providecommand</code> 37, 449, 450, 626, 627, 638, 953, 955
<code>\ProvidesFile</code>	3
<code>\ProvidesPackage</code>	631
R	
<code>\RaggedRight</code>	681
<code>\raggedright</code>	680
<code>\RawNoindent</code>	853
<code>\RawParEnd</code>	847
<code>\ref</code>	23, 421
ref commands:	
<code>\ref_attribute_gset:nnnn</code>	462
<code>\ref_label:nn</code>	469
<code>\ref_value:nn</code>	80, 514, 516
<code>\relax</code>	191, 290, 302, 642, 643, 704, 707, 717, 724, 750, 758, 763, 765, 766, 768, 892, 920, 921, 962, 965, 972, 976, 1025, 1060, 1094, 1193
<code>\renewcommand</code>	647, 650, 680, 681, 716
<code>\RequirePackage</code>	868
<code>\rightmargin</code>	912
<code>\rule</code>	23, 295, 854
S	
seq commands:	
<code>\seq_clear:N</code>	488
<code>\seq_count:N</code>	397
<code>\seq_map_inline:Nn</code>	565, 585
<code>\seq_new:N</code>	75
<code>\seq_put_right:Nn</code>	494
<code>\seq_set_split:Nnn</code>	396
<code>\seq_show:N</code>	584
<code>\setbox</code>	13, 286, 430, 744, 849, 904
<code>\setfnsymbol</code>	652, 1010, 1161
<code>\setlength</code>	318, 660
<code>side (plug)</code>	5
<code>\skip</code>	717
socket commands:	
<code>\socket_log:n</code>	136, 137, 138, 139, 140, 141, 142, 143, 144, 145, 146
<code>\socket_use:n</code> ..	207, 253, 266, 272, 323
<code>\socket_use:nn</code> ..	179, 208, 248, 323, 324
Sockets:	
<code>fntext/begin</code>	5, 5, 294
<code>fntext/end</code>	5, 5, 297
<code>fntext/make</code>	5, 5, 5, 5, 5, 6, 282
<code>fntext/mark</code>	5, 5, 326
<code>fntext/process</code>	5, 5, 278
<code>fntext/text</code>	6, 6, 343
<code>tagssupport/fnmark</code> ...	7, 7, 10, 27, 199
<code>tagssupport/fntext/begin</code> .	7, 7, 10, 306
<code>tagssupport/fntext/end</code> ..	7, 7, 10, 306
<code>tagssupport/fntext/mark</code>	7, 7, 10, 29, 344

`\tag`support/fntext/text [8](#), [8](#), [10](#), [30](#), [344](#)
`\space` [4](#)
`\spacefactor` [165](#), [191](#), [967](#), [972](#)
`\splitfootnoterule` [723](#), [731](#)
`\splitmaxdepth` [226](#)
`\splittopskip` [225](#)
`\stepcounter` [35](#), [87](#), [871](#), [878](#), [1186](#)
`\strut` [301](#)
`\strutbox` [23](#), [226](#), [298](#)

T

tag commands:

`\tag_check_child:nnTF` [553](#)
`\tag_get:n` [481](#), [507](#), [513](#), [560](#)
`\tag_if_active:TF` [537](#)
`\tag_mc_begin:n` ... [518](#), [588](#), [598](#), [955](#)
`\tag_mc_begin_pop:n`
 [532](#), [577](#), [592](#), [601](#), [955](#)
`\tag_mc_end:` [530](#), [590](#), [600](#), [955](#)
`\tag_mc_end_push:`
 [502](#), [552](#), [583](#), [597](#), [955](#)
`\tag_struct_begin:n`
 [503](#), [555](#), [558](#), [561](#), [587](#)
`\tag_struct_end:` ... [531](#), [561](#), [576](#), [591](#)
`\tag_struct_gput:nnn` [539](#)
`\tag_struct_object_ref:n` [539](#)
`\tag_struct_use:n` [83](#)
`\tag`support/fnmark (socket) [7](#), [7](#), [10](#), [27](#), [199](#)
`\tag`support/fntext/begin (socket)
 [7](#), [7](#), [10](#), [306](#)
`\tag`support/fntext/end (socket) [7](#), [7](#), [10](#), [306](#)
`\tag`support/fntext/mark (socket)
 [7](#), [7](#), [10](#), [29](#), [344](#)
`\tag`support/fntext/text (socket)
 [8](#), [8](#), [10](#), [30](#), [344](#)

TeX and L^AT_EX 2_ε commands:

`\@@dvance@macro` [641](#), [642](#)
`\@@par` [917](#)
`\@MM` [227](#)
`\@advance@macro` [646](#)
`\@arabic` [1087](#)
`\@auxout` [639](#)
`\@bsphack` [1011](#), [1088](#)
`\@ctrerr` [1045](#), [1072](#), [1092](#)
`\@currentcounter` [18](#), [231](#)
`\@currentlabel` [19](#), [232](#)
`\@diagnose@fnsymbol@orange`
 [1094](#), [1095](#), [1097](#)
`\@eha` [1015](#)
`\@empty` [450](#), [677](#)
`\@esphack` [1020](#), [1091](#)
`\@finalstrut` [23](#), [298](#)
`\@fnsymbol` [650](#), [1017](#), [1022](#)
`\@fnsymbol@orange` [1090](#), [1096](#)

`\@footnotemark` [3](#), [16](#),
 [23](#), [200](#), [408](#), [416](#), [425](#), [439](#), [1188](#), [1196](#)
`\@footnotetext` [3](#), [17](#), [310](#), [402](#), [410](#), [444](#)
`\@gobble` [991](#)
`\@gobbletwo` [449](#)
`\@ifclassloaded` [797](#)
`\@ifnextchar` [989](#), [1004](#), [1183](#)
`\@ifpackageloaded` [454](#), [794](#), [860](#)
`\@ifundefined` [679](#), [1012](#)
`\@listdepth` [725](#)
`\@makecol` [436](#), [441](#)
`\@makefnmark` [3–5](#), [7](#), [10](#), [16](#),
 [21](#), [34](#), [179](#), [329](#), [332](#), [335](#), [337](#), [349](#),
 [362](#), [377](#), [378](#), [906](#), [908](#), [935](#), [938](#), [940](#)
`\@makefntext` [3](#), [5](#), [6](#),
 [10](#), [11](#), [20–22](#), [27](#), [22](#), [230](#), [244](#), [283](#),
 [286](#), [355](#), [370](#), [383](#), [385](#), [440](#), [902](#), [931](#)
`\@makefntext@processX`
 [374](#), [376](#), [378](#), [379](#)
`\@mpfn` [29](#), [401](#), [406](#), [1186](#), [1193](#)
`\@mpfootins` [13](#), [14](#), [430](#), [431](#)
`\@mpfootnotetext` [9](#), [12](#)
`\@mplistdepth` [725](#)
`\@ne` [693](#), [862](#), [915](#)
`\@null` [1067](#)
`\@outputbox@append` [751](#), [756](#)
`\@outputbox@reinsertbskip`
 [755](#), [758](#), [763](#), [766](#)
`\@outputbox@removebskip` [749](#), [765](#)
`\@parboxrestore` [17](#), [229](#), [844](#)
`\@plus` [717](#)
`\@setpar` [917](#)
`\@stpelt` [870](#)
`\@tempa` [1052](#), [1053](#), [1078](#), [1079](#)
`\@tempb` [1031](#), [1053](#), [1066](#), [1079](#)
`\@tempboxa` [825](#), [904](#), [911](#), [919](#)
`\@tempc` [1033](#), [1036](#)
`\@tempcnta` [642](#), [643](#), [644](#)
`\@tempd` [1034](#), [1036](#)
`\@tempskipa` [752](#), [753](#), [754](#), [756](#)
`\@textbottom` [750](#)
`\@textsuperscript` [34](#)
`\@thefnmark` [8–10](#), [12](#),
 [14](#), [23](#), [26](#), [27](#), [29](#), [20](#), [34](#), [88](#), [93](#),
 [100](#), [105](#), [232](#), [421](#), [506](#), [564](#), [1187](#), [1194](#)
`\@typeset@protect` [980](#), [995](#)
`\@x@sf` [165](#), [191](#), [967](#), [972](#)
`\@xmpfootnotemark` [1184](#), [1191](#)
`\iiiminipage` [10](#)
`\c@footnote` [650](#)
`\color@begingroup` [21](#), [242](#)
`\color@endgroup` [25](#), [269](#)
`\doublecol@number` [862](#)
`\FN@abovefloatsfalse` [694](#), [702](#)

<code>\FN@abovefloatstrue</code> . . .	691, 699, 701	<code>\l@advance@macro</code>	641, 646
<code>\FN@baselinestretch</code>	800, 804	<code>\m@m@singlespace</code>	799, 800
<code>\FN@bottomcases</code> . . .	690, 693, 698, 768	<code>\m@ne</code>	870, 876
<code>\FN@build@symboldef</code> .	1032, 1051, 1056	<code>\new@std@class@makecol</code>	436, 441
<code>\FN@build@symboldefTM</code>	1067, 1077, 1082	<code>\nfss@text</code>	1039
<code>\FN@fixskipfalse</code>	688	<code>\old@std@class@makefntext</code>	346, 383, 440
<code>\FN@fixskiptrue</code> . . .	695, 699, 701, 702	<code>\p@</code>	717, 723
<code>\FN@fnsymbol@lamport</code>	1022	<code>\p@footnote</code>	232
<code>\FN@hangfootfalse</code>	709	<code>\pp@cl@end@iii</code>	878
<code>\FN@hangfoottrue</code>	711	<code>\pp@fix@MakePerPage</code> . . .	872, 874, 878
<code>\FN@makefootnoteparagraph</code> . .	744, 842	<code>\protected@edef</code>	19, 232
<code>\FN@mf@check</code>	957, 964	<code>\protected@write</code>	639
<code>\FN@mf@prepare</code>	958, 960, 976	<code>\protected@writeaux</code>	638
<code>\FN@multiplefootnotefalse</code>	739	<code>\protected@xdef</code>	88, 100, 1187
<code>\FN@multiplefootnotetrue</code>	740	<code>\reset@font</code>	15, 217
<code>\FN@orange</code>	1046, 1073, 1085	<code>\setspace@singlespace</code> .	445, 803, 804
<code>\FN@parafalse</code>	654	<code>\split@prev</code>	720, 728, 733
<code>\FN@paratrue</code>	659	<code>\thr@@</code>	690
<code>\FN@perpagefalse</code>	684	<code>\toks@</code> 1030, 1043, 1055, 1065, 1070, 1081	
<code>\FN@perpagetrue</code>	686	<code>\tw@</code>	698
<code>\FN@removehboxes</code>	846, 849, 850	<code>\unrestored@protected@xdef</code>	93, 105, 421, 1194
<code>\FN@robustfalse</code>	648	<code>\z@</code> .	23, 287, 295, 328, 331, 669, 728, 753, 854, 875, 890, 905, 912, 934, 937
<code>\FN@robusttrue</code>	651	<code>\textasteriskcentered</code>	1106, 1114, 1122, 1123, 1131, 1137, 1142, 1148, 1153, 1163, 1169, 1174
<code>\FN@setfootnoteparawidth</code> 843, 861, 865		<code>\textbardbl</code>	1110, 1118, 1128, 1136, 1147, 1168
<code>\FN@setspacefalse</code>	808	<code>\textdagger</code> 1107, 1115, 1124, 1132, 1138, 1143, 1149, 1154, 1164, 1170, 1175	
<code>\FN@setspacetrue</code>	796	<code>\textdaggerdbl</code>	1108, 1116, 1125, 1133, 1139, 1144, 1150, 1155, 1165, 1171, 1176
<code>\FN@sf@@footnote</code>	979, 981	<code>\TextOrMath</code>	1081
<code>\FN@sf@@footnotemark</code>	994, 996	<code>\textparagraph</code>	1111, 1127, 1135, 1146, 1152, 1158, 1167, 1173, 1179
<code>\FN@sf@gobble@bracket</code> . . .	1005, 1008	<code>\textsection</code> . .	1109, 1117, 1126, 1134, 1145, 1151, 1156, 1166, 1172, 1177
<code>\FN@sf@gobble@opt</code>	983, 986	<code>\textsuperscript</code>	971
<code>\FN@sf@gobble@optonly</code>	998, 1001	<code>\textwidth</code>	863
<code>\FN@sf@gobble@twobracket</code> . . .	990, 993	<code>\the</code>	165, 644, 733, 756, 967, 1043, 1055, 1070, 1081
<code>\FN@stablefootnotefalse</code>	737	<code>\thefootnote</code>	30, 647, 650
<code>\FN@stablefootnotetrue</code>	738	<code>\thempfn</code>	30, 1187, 1194
<code>\FN@tempboxa</code>	286, 287, 288, 289, 291, 825, 849, 850, 851	tl commands:	
<code>\FN@tempboxb</code>	826	<code>\tl_if_eq:nTF</code>	492
<code>\FN@tempboxc</code>	827	<code>\tl_if_in:NnTF</code>	356, 359, 362
<code>\FN@temptoken</code>	637	<code>\tl_if_novalue:nTF</code>	85, 98
<code>\hb@xt@</code>	21, 329, 337, 349, 359, 375, 376, 906, 935, 940	<code>\tl_new:N</code> 66, 67, 71, 73, 74, 78, 350, 548	
<code>\hyper@linkend</code>	450, 527	<code>\tl_set:Nn</code>	68, 72, 79, 355, 368, 424, 507, 516, 549, 560
<code>\hyper@linkstart</code>	449, 523		
<code>\hyper@nopatch@footnote</code>	446		
<code>\if@tempswb</code>	1023		
<code>\ifFN@abovefloats</code> . .	691, 771, 777, 783		
<code>\ifFN@fixskip</code>	688, 748		
<code>\ifFN@hangfoot</code>	709, 901		
<code>\ifFN@multiplefootnote</code>	739, 952		
<code>\ifFN@para</code> .	654, 662, 742, 813, 824, 882		
<code>\ifFN@perpage</code>	683, 867		
<code>\ifFN@robust</code>	648, 1086		
<code>\ifFN@setspace</code>	793		
<code>\ifFN@stablefootnote</code>	737, 978		

U		V	
\unhbox	851	\value	875
\unkern	969, 970	\vbox	13, 430, 744
\unskip	160	\vskip	754, 756
\unvbox	14, 431, 846		
use commands:		W	
\use:N	88, 93, 100, 105	\wd	289, 911
\use:n	378		
\UseHook	188, 195	X	
\UseSocket	355, 379	\xdef	733, 755, 1035, 1068