

The `latex-lab-footnotes` code*

Frank Mittelbach, L^AT_EX Project

May 25, 2023

Abstract

to be written

Contents

1	Introduction	2
1.1	Configuration methods	3
2	Hooks and configuration points	3
2.1	Formatting the mark in the main text	3
2.1.1	Configuration points	3
2.1.2	Hooks	3
2.1.3	Additional configuration possibilities	4
2.2	Formatting the footnote text	4
2.2.1	Configuration points	4
2.2.2	Hooks	5
2.2.3	Additional configuration possibilities	6
3	Tagging and hyperlinking support	6
3.1	Technical details for the tagging	7
3.2	Requirements for links	7
3.3	The algorithmus to connect marks and notes	7
3.3.1	<code>\footref</code>	8
3.3.2	<code>\footnotemark</code> after <code>\footnotetext</code>	8
3.4	Links	9
3.5	Implementation details	9
3.6	Handling the mark	9
3.7	Handling the <code>footnotetext</code>	9
3.8	Footnotes in minipages	9
4	TODOs	9

*

5	The Implementation	10
5.1	File declaration	10
5.2	code not fully handled yet	10
5.3	Temporary variables	11
5.4	Public variables	11
5.5	Internal variables	12
5.6	Variants	13
5.7	Updating <code>\@thefnmark</code>	13
5.8	Hooks	13
5.9	Debugging code	14
5.10	The new <code>\@footnotemark</code> command	15
5.11	The new <code>\@footnotetext</code> command	16
5.12	The new <code>\@makefnmark</code> command	18
5.12.1	Making documents use the new <code>\@makefnmark</code>	19
5.13	Document-level commands	21
5.14	Firstaid for packages and classes	22
5.15	Kernel patches	22
5.15.1	memoir	22
5.15.2	setspace	22
5.15.3	hyperref	23
5.16	Tagging and hyperlink code	23
5.16.1	Rolemap for structure tags	23
5.16.2	Extending the label system	23
5.16.3	Storing and retrieving reference data	24
5.16.4	Enabling tagging and links for the mark command	25
5.16.5	The footnote text	26
6	Reimplementing the footmisc package	29
	Index	41

1 Introduction

This code reimplements the footnote interfaces for \LaTeX offering configurable methods for layout and functionality adjustments that avoid overwriting each other when used in classes as well as in packages (as far as possible — obviously some adjustments are mutually exclusive). This is achieved by providing a larger number of hooks (for areas where different packages/classes can easily coexist with their adjustments) and a number of configuration points 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 if will not work in a 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 Hooks and configuration points

Note: the configuration points do not have an interface mechanism yet and all their names are temporary right now. Also note that configuration points 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 Configuration points

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

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

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`.

¹Like this one.

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 `\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`, `scrlltr2`, 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 Configuration points

To cater for different layout configurations there are four configuration points which can be set only by one package or class, if two packages/classes set them they are mutually incompatible. These are:

`\@footnotetext@cfgpoint` (1 argument) This receives all material that is to be processed (or stored) including color protection code and what have you. The default definition is to run `\insert\footins`.

`\@footnotetext@cfgpointii` (1 argument) The default definition runs `\@makefnmark` 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 in the insert and this configuration point allows running extra code to do that.

`\@footnotetext@cfgpointiii` (no argument) By default this configuration point 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 this configuration point can cancel the action or do something else instead.

The configuration point is executed near the start of the argument for the configuration point `\@footnotetext@cfgpointii`.

`\@footnotetext@cfgpointiv` (**no argument**) This configuration point is executed at the very end of the argument passed to `\@footnotetext@cfgpointii`. 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.

The configuration point `\@footnotetext@cfgpointii` runs `\@makefntext` and this command contains two further configuration points (and a few hooks):

`\@makefntext@cfgpoint` (**1 argument**) This configuration point receives the material to typeset the footnote mark. By default, all it does is running `\indent` to get a paragraph indentation (if one is set up—in most layouts it is 0 points) and then typesets the mark, but in some designs it executes more elaborate code.

If tagging is produced this configuration point is also responsible for surrounding the mark with the appropriate tags marking the mark as an Lbl. It does this using the command `\tag@FENoteLb1`.

`\@makefntext@cfgpointii` (**1 argument**) This configuration point manages the formatting of the footnote text once the mark has been typeset.

If tagging is produced this configuration point is also responsible for surrounding the mark with the appropriate tags marking the mark as an MC of type FENote. It does this using the command `\tag@FENotetext`.

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

fntext/before Executed at the very beginning of `\footnotetext`. Currently there is on 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 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. 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.

The configuration point `\@footnotetext@cfgpointii` normally runs the command `\@makefntext` or some code that eventually runs this command, and this then produces the footnote mark (in front of the footnote text) and 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 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

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 *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 data structure

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.

and to switch locally to use this container. The following three commands are used for this.

The default property uses the name `default`

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

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

<code>\fnote_mark_gput:nn</code>	<code>\fnote_mark_gput:nn{<mark>}{<footnote type name>}</code>
----------------------------------	--

`\fnote_mark_gput:(no|oo)`

This command stores the current structure number as key and the *<mark>* as value in the property associated with the *<footnote type name>*

<code>\fnote_mark_gpop:nnN</code>	<code>\fnote_mark_gpop:nnN{<mark>}{<footnote type name>}{<sequence>}</code>
-----------------------------------	---

This command stores the keys/structure numbers whose value are *<mark>* in the property associated with *<footnote type name>* in the *<sequence>* and then remove them from the property. 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 (currently with the help of a hook from the `nameref` package) 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

3.6 Handling the mark

The mark in the text is handled by redefining the kernel configuration point `\@kernel@process@makefnmark` to `\tag@FEMark`. It takes one argument, `\@makefnmark`, the command which formats the mark, and surrounds it by link and tagging commands. At the point where `\@kernel@process@makefnmark` is issued `\@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 redefining `\@kernel@process@footnotetext`. This configuration point takes two arguments, `\@footnotetext@cfigpoint` (by default `\insert\footins`) and as second argument lots of code related to typesetting the notemark and the footnote text with the actual content of the footnote text somewhere in the middle.

The redefinition of `\@kernel@process@footnotetext` surrounds the content with the structure command and tries to detect to which mark the note is related.

The actual typesetting of the note text is done by `\@makefntext/\fnote_-makefntext:n`. In the new implementation this contains two configuration points, `\@makefntext@cfigpoint` and `\@makefntext@cfigpointii`. These are redefined to 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 on `nameref` as it provides the hook in `\label` used by the `\footref` code.

- 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.
- Check if additional kernel configuration points are needed/possible to avoid the redefinitions of `\@makefnmark` and `\@makefnmarkii`.
- 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 [2023-02-04 v0.7 changes to the footnote interfaces]
```

5.2 code not fully handled yet

```
5 %
6 % latex.ltx
7 % not looked at yet
8 % \mpfootnotetext is probably no longer needed, or only to support other
9 % classes and package. See below about the minipage code.
10 %
11 % \long\def\@mpfootnotetext#1{%
12 %   \global\setbox\@mpfootins\vbox{%
13 %     \unvbox\@mpfootins
14 %     \reset@font\footnotesize
15 %     \hsize\columnwidth
16 %     \@parboxrestore
17 %     \def\@currentcounter{mpfootnote}%
18 %     \protected@edef\@currentlabel
19 %       {\csname p@mpfootnote\endcsname\@thefnmark}%
20 %     \color@begingroup
21 %       \@makefnmark{%
22 %         \rule{z@footnotesep\ignorespaces#1\@finalstrut\strutbox}%
23 %       }
24 %     \color@endgroup}}
25 % =====
26 % used by the minipage footnote code.
27 %
28 % \def\@mpfn{footnote}
29 % \def\thempfn{\thefootnote}
```

```

30 % =====
31 % this perhaps need some configuration options.
32 %
33 %\def\@makefnmark{\hbox{\@textsuperscript{\normalfont\@thefnmark}}}%
34 %
35 % =====
36 %% alterations not covered:
37 %
38 % ./arabtex/afoot.sty --- too different (and probably too old)
39 %
40 % =====
41 % alterations of footnotetext not covered:
42 %
43 % ./revtex4-1/revtex4-1.cls ./revtex/ltxutil.sty ./revtex/revtex4-2.cls ... (need analysis)
44 % ./bigfoot/bigfoot.sty
45 %
46 % memoir needs checking too
47 %
48 % =====
49 %
50 % use of kerns to mark h-mode positions (unit sp)
51 %
52 % 1 = CJK
53 % 2 = CJK
54 % 3 = multiple footnotes (footmisc, koma, eledmac, tufte, memoir,
55 %   parnotes, sidenotes)
56 % 3 = outer kern in letter spacing (letterspace)
57 % 3 = beginning of list (examdesign.cls)
58 % 4 = CJK pigin
59 % 5 = CJK ruby
60
61 % 1-4 = polyglossia for korean
62 %

```

```

63 \ExplSyntaxOn

```

5.3 Temporary variables

```

64 \prop_new:N \l__fnote_tmpa_prop
65 \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

```

```

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

```

(End 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`

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

(End 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`

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

(End 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

```
72 \tl_new:N \l__fnote_linktarget_tl
```

(End definition for \l__fnote_linktarget_tl.)

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

```
73 \tl_new:N \l__fnote_currentlabel_tl
```

(End definition for \l__fnote_currentlabel_tl.)

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

```
74 \seq_new:N \l__fnote_currentrefs_seq
```

(End 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`

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

(End 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.

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

5.6 Variants

```

79 \cs_generate_variant:Nn \ref_label:nn { Vn }
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 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 definition for \fnote_set_fnmark:nn. This function is documented on page ??.)

5.8 Hooks

Hooks in the footnotemark command.

```

109 \NewMirroredHookPair{fnmark/before}{fnmark/after}
110 \NewHook{fnmark}

```

```

111 \NewHook{fnmark/begin}
112 \NewHook{fnmark/end}

```

Hooks in the footnotetext command

```

113 \NewMirroredHookPair{fntext/before}{fntext/after}
114 \NewHook{fntext}
115 \NewHook{fntext/para}
116 \NewHook{fntext/begin}
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     \LogHook{fnmark/before}
125     \LogHook{fnmark}
126     \LogHook{fnmark/begin}
127     \LogHook{fnmark/end}
128     \LogHook{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     \cs_log:N\@footnotetext@cfgpoint
137     \cs_log:N\@footnotetext@cfgpointii
138     \cs_log:N\@footnotetext@cfgpointiii
139     \cs_log:N\@footnotetext@cfgpointiv
140     \cs_log:N\@makefntext@cfgpoint
141     \cs_log:N\@makefntext@cfgpointii
142     \LogHook{fntext/before}
143     \LogHook{fntext}
144     \LogHook{fntext/para}
145     \LogHook{fntext/begin}
146     \LogHook{fntext/end}
147     \LogHook{fntext/after}

```

Show the info only once (if at all).

```

148     \cs_gset_eq:NN \__fnote_debug_footnotetext: \prg_do_nothing:
149   }
150 }

```

5.10 The new \@footnotemark command

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

```

151 \cs_new_protected:Npn \fnote_footnotemark: {
152   \__fnote_debug_footnotemark:
153   %-----
154   % bibarts
155   % chextras --- actually in the wrong place does an \unskip
156   \UseHook{fnmark/before}
157   %-----
158   \leavevmode
159   \ifhmode
160     \edef\x@sf{\the\spacefactor}
161   %-----
162   % bxjsja-minimal.def --- what they do could be done at ‘‘bibarts’’
163   %                               (a bit less efficient)
164   % memhfixc.sty
165   % footmisc.sty
166   \UseHook{fnmark}
167   %-----
168   \nobreak
169   \fi
170   %-----
171   % hyperref.sty
172   \UseHook{fnmark/begin}
173   %-----
174   \@kernel@process@makefnmark
175   \@makefnmark
176   %-----

```

If a footnote mark is placed by its own then it should finish by executing `fnmark/end`, resetting the space factor, and finishing with `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.

```

177   \__fnote_footnotemark_finish:
178 }

```

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

```

\__fnote_finish:
\__fnote_footnotemark_finish:
179 \cs_new_protected:Npn \__fnote_finish: {
180 % hyperref.sty
181 % memhfixc.sty --- could move fnmark/after
182 % scr1ttr2.cls --- could vanish if footmisc uses a hook
183 % footmisc.sty
184 \UseHook{fnmark/end}
185 %-----
186 \ifhmode
187   \spacefactor \@x@sf \relax
188 \fi
189 %
190 %-----
191 \UseHook{fnmark/after}

```

```

192 %-----
193 }
194
195 \cs_new_eq:NN \__fnote_footnotemark_finish: \__fnote_finish:

```

(End definition for __fnote_finish: and __fnote_footnotemark_finish:.)

```

\@kernel@process@makefnmark Not a public config point but the kernel hook to add tagging
196 \def \@kernel@process@makefnmark { }

```

(End definition for \@kernel@process@makefnmark. This function is documented on page ??.)

\@footnotemark At last provide the name $\text{\LaTeX} 2_{\epsilon}$ is used to.

```

197 \cs_set_eq:NN \@footnotemark \fnote_footnotemark:

```

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

5.11 The new \fnotetext command

\fnote_footnotetext:n

```

198 \cs_new_protected:Npn \fnote_footnotetext:n #1 {
199   \__fnote_debug_footnotetext:
200   %-----
201   % ./linguex/linguex.sty
202   \UseHook{fntext/before}
203   %-----
204   \@kernel@process@footnotetext
205   \@footnotetext@cfgpoint { % config point
206   %-----
207   % resetting baselinestretch ... (could be done further down)
208   % ./uafthesis/uafthesis.cls
209   % ./setspace/setspace.sty
210   % ./footmisc/footmisc.sty (normal)
211   \UseHook{fntext}
212   %-----
213   \reset@font
214   \footnotesize
215   %-----
216   % some classes use a different font size, e.g.,
217   % ./nrc/nrc1.cls ./nrc/nrc2.cls
218   % but those could be done in fntext/para instead
219   %-----

```

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

```

220   \interlinepenalty\interfootnotelinepenalty
221   \splittopskip\footnotesep
222   \splitmaxdepth \dp\strutbox
223   \floatingpenalty \@MM
224   \hspace\columnwidth
225   \@parboxrestore
226   \parindent 1em % typical default used in \@makefntext moved up here
227   \def\@currentcounter{footnote}
228   \protected@edef \@currentlabel { \p@footnote \@thefnmark }

```



```

229 %-----
230 % for altering para parameters ...
231 % code for resphilosophica came earlier but it could go here.
232 % Has the advantage that one can also overwrite \cs{@currentcounter}
233 % and \cs{@currentlabel} is that is necessary.
234 %
235 % ./resphilosophica/resphilosophica.cls
236 \UseHook{fntext/para}
237 %-----
238 \color@begingroup
239 %-----
240 % fnpara wants to replace \@makefntext{...} and para and side option of footmisc etc too...
241 % so we make this a config point
242 %-----
243 \@footnotetext@cfgpointii % config point
244 {
245 %-----
246 % ./resphilosophica/resphilosophica.cls
247 %-----
248 \@footnotetext@cfgpointiii % config point
249 %-----
250 % bibarts
251 % fnbreak.sty
252 \UseHook{fntext/begin}
253 %-----
254 \ignorespaces
255 #1
256 %-----
257 % bibarts
258 % fnbreak.sty
259 \UseHook{fntext/end}
260 %-----
261 \@footnotetext@cfgpointiv % config point
262 }
263 \par
264 \color@endgroup
265 }
266 %-----
267 % ./linguex/linguex.sty
268 \UseHook{fntext/after}
269 %-----
270 }

```

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

```

\@footnotetext@cfgpoint default for config point (1 arg)
271 \cs_new_protected:Npn \@footnotetext@cfgpoint { \insert\footins }

```

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

```

\@footnotetext@cfgpointii default for config point (1 arg)
272 \cs_new_protected:Npn \@footnotetext@cfgpointii { \@makefntext }

```

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

```
\@footnotetext@cfgpointiii default for config point (0 args)
273 \cs_new_protected:Npn \@footnotetext@cfgpointiii { \rule{z@}{\footnotesep} }
(End definition for \@footnotetext@cfgpointiii. This function is documented on page ??.)
```

```
\@footnotetext@cfgpointiv default for config point (0 args)
274 \cs_new_protected:Npn \@footnotetext@cfgpointiv { \@finalstrut\strutbox }
(End definition for \@footnotetext@cfgpointiv. This function is documented on page ??.)
```

```
\@kernel@process@footnotetext kernel hook for tagging (2 args)
275 \cs_new_protected:Npn \@kernel@process@footnotetext {}
(End definition for \@kernel@process@footnotetext. This function is documented on page ??.)
```

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).

```
276 \AddToHook{begindocument}
277 {
278   \cs_set_eq:NN \@footnotetext \fnote_footnotetext:n
279 }
```

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.

```
280 \newdimen\footnotemargin
281 \footnotemargin\maxdimen % no value given
282
283 \AtBeginDocument
284 {
285   \ifdim \footnotemargin=\maxdimen
286     \setlength\footnotemargin{1.8em}
287   \fi
288 }
```

```
\fnote_makefntext:n
```

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

Some classes in their redefinition for `\@makefntext` have places 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.

```
290   \@makefntext@cfgpoint
291   {
292     \ifdim\footnotemargin>\z@
293       \hb@xt@ \footnotemargin{\hss\@makefnmark}
294     \else
295       \ifdim\footnotemargin=\z@
296         \llap{\@makefnmark}
297       \else
298         \ifdim\footnotemargin=-\maxdimen
299           \@makefnmark
300         \else
301           \llap{\hb@xt@ -\footnotemargin{\@makefnmark\hss}}
```

```

302         \fi
303     \fi
304     \fi
305 }
306 \@makefntext@cfgpointii
307 { #1 }
308 }

```

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

`\@makefntext@cfgpoint` default for config point (1 arg)

```

309 \cs_new_protected:Npn \@makefntext@cfgpoint { \noindent }

```

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

`\@makefntext@cfgpointii` default for config point (1 arg)

```

310 \cs_new_protected:Npn \@makefntext@cfgpointii #1 { #1 }

```

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

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.

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`

```

311 \newcommand\old@std@class@makefntext[1]{%
312     \parindent 1em%
313     \noindent
314     \hb@xt@1.8em{\hss\@makefnmark}#1}

```

(End 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 configuration points in the right place.

What it does is roughly the following: It look 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 \@makefntext@cfgpoint{\hbox BBB { CCC }} DDD
  \@makefntext@cfgpointii{#1} EEE }
```

The patching is not very careful, i.e., it assumes there is only one `#1` in the replacement text and that a `\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).

```

315 \tl_new:N \l__fnote_patch_tl
316 \cs_new_eq:NN \__fnote_tmp:w \ERROR
317
318 \cs_new_protected:Npn \__fnote_patch:
319 {
320   \tl_set:No \l__fnote_patch_tl { \@makefntext { \@makefntext@cfgpointii{##1} } }
321   \tl_if_in:NnTF \l__fnote_patch_tl { \hbox }
322     { \cs_set_eq:NN \__fnote_tmp:w \__fnote_patch_hbox:w }
323     {
324       \tl_if_in:NnTF \l__fnote_patch_tl { \hb@xt@ }
325         { \cs_set_eq:NN \__fnote_tmp:w \__fnote_patch_hb@xt@:w }
326         {
327           \tl_if_in:NnTF \l__fnote_patch_tl { \@makefnmark }
328             { \cs_set_eq:NN \__fnote_tmp:w \__fnote_patch_@makefnmark:w }
329             { \ERROR
330               \cs_set_eq:NN \__fnote_tmp:w \exp_stop_f: }
331           }
332         }
333   \tl_set:Nf \l__fnote_patch_tl
334     { \exp_after:wN \__fnote_tmp:w \l__fnote_patch_tl }
335   \cs_set:Npn \__fnote_tmp:w { \long \def \@makefntext ###1 }
336   \exp_after:wN \__fnote_tmp:w \exp_after:wN { \l__fnote_patch_tl }
337 }

```

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

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

```

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

```

Same for the other cases.

```

340 \cs_new:Npn \__fnote_patch_hb@xt@:w #1 \hb@xt@ #2 #
341 { \exp_stop_f: #1 \@makefntext@processX { \hb@xt@ #2 } }
342 \cs_new:Npn \__fnote_patch_@makefnmark:w #1 \@makefnmark
343 { \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:

```
344 \cs_new:Npn \@makefntext@processX #1#2{\@makefntext@cfgpoint{#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 using the approach above.

```

345
346 \AddToHook{begindocument}
347 {
348   \cs_if_eq:NNTF \@makefntext \old@std@class@makefntext
349     {
350       \cs_set_eq:NN \@makefntext \fnote_makefntext:n
351     }

```

```

352     {
353       \__fnote_patch:
354     }
355   }
356
357
358   % possibly add the following to check for multiple \hbox in
359   % the definition:
360   %
361   % \seq_set_split:NnV \l__fnote_patch_seq { \hbox } \l__fnote_patch_tl
362   % \int_compare:nT { \seq_count:N \l__fnote_patch_seq } > 2 \ERROR
363   %

```

5.13 Document-level commands

`\footnotetext`

```

364 \DeclareDocumentCommand\footnotetext {o+m}
365 {
366   \fnote_set_fnmark:nn {#1} \@mpfn
367   \@footnotetext {#2}
368 }

```

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

`\footnote`

```

369 \DeclareDocumentCommand\footnote {o+m}
370 {
371   \fnote_step_fnmark:nn {#1} \@mpfn
372   \cs_set_eq:NN \__fnote_footnotemark_finish: \prg_do_nothing:
373   \@footnotemark
374   \cs_set_eq:NN \__fnote_footnotemark_finish: \__fnote_finish:
375   \@footnotetext {#2}
376   \__fnote_footnotemark_finish:
377 }

```

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

`\footnotemark`

```

378 \DeclareDocumentCommand\footnotemark {o}
379 {
380   \fnote_step_fnmark:nn {#1} { footnote }
381   \@footnotemark
382 }

```

(End 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 related note automatically but should instead use the label. This is passed to `\@footnotemark` through `\l__fnote_currentlabel_tl`.

```

383 \DeclareDocumentCommand\footref {m}
384 {
385   \begingroup
386   \unrestored@protected@xdef\@thefnmark{\ref*{#1}}%

```

```

387 \endgroup
388 \bool_set_false:N \l__fnote_autodetect_bool
389 \tl_set:Nn \l__fnote_currentlabel_tl {#1}
390 \@footnotemark
391 \bool_set_true:N \l__fnote_autodetect_bool
392 }

```

(End 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.

TODO remove the dependency to etoolbox, and need for the patch.

```

393 \cs_new_protected:Npn \@mpfootnotetext@cfgpoint #1
394 {
395   \global\setbox\@mpfootins\vbox{%
396     \unvbox\@mpfootins
397     #1
398   }
399 }

```

Then we patch `\@iiminipage`

```

400 \RequirePackage{etoolbox}
401 \ExplSyntaxOff
402 \patchcmd\@iiminipage
403   {\let\@footnotetext\@mpfootnotetext}
404   {\let\@footnotetext\@cfgpoint\@mpfootnotetext@cfgpoint}
405   {}{\ERROR}
406 \ExplSyntaxOn

```

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.

```

407 \AddToHook{class/memoir/before}
408 { \let\new@std@class@makecol\@makecol }
409 \AddToHook{class/memoir/after}
410 {
411   \cs_set_eq:NN \@footnotemark \fnote_footnotemark:
412   \cs_set_eq:NN \@makefntext\old@std@class@makefntext
413   \cs_set_eq:NN \@makecol\new@std@class@makecol
414 }

```

5.15.2 setspace

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

```

415 \AddToHook{package/setspace/after}
416   {\let \@footnotetext \fnote_footnotetext:n
417     \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.

```
418 \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.

```
419 \AtBeginDocument
420 {
421   \providecommand\hyper@linkstart{\@gobbletwo}
422   \providecommand\hyper@linkend{\@empty}
423 }
```

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).

```
424 \AtBeginDocument
425 {
426   \ifpackageloaded{hyperref}
427   {
428     \legacy_if:nF{Hy@hyperfootnotes}{\bool_set_false:N \l_fnote_link_bool}
429   }
430   {
431     \bool_set_false:N \l_fnote_link_bool
432   }
433 }
```

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 `nameref` hook until latex provides a hook for its label. At first we define a suitable attribute, it uses as value the structure number of the note as stored in `\l__fnote_currentstruct_tl`

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

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

```
\__fnote_label_hook:
```

```
435 \cs_new_protected:Npn \__fnote_label_hook: {}
436 \AddToHook{begindocument/before}
437 {
438   \RequirePackage{nameref}
439   \g@addto@macro\label@hook
440   {
441     \__fnote_label_hook:
```

```

442     }
443 }

```

(End definition for `_fnote_label_hook:`.)

Inside a `footnotetext` we change the hook to store the structure number too. `nameref` provides the name of label in `\label@name`.

```

444 \AddToHook{fntext/begin}
445 {
446   \cs_set_protected:Npn \_fnote_label_hook:
447   {
448     \ref_label:Nn \label@name {fnote/struct}
449   }
450 }

```

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_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!

```

451 \cs_new_protected:Npn \fnote_new:nn #1 #2 % #1 name, #2 options
452 {
453   \prop_new:c { g__fnote_currentmarks_ #1 _prop }
454 }
455
456 \fnote_new:nn {default}{}

```

(End definition for `\fnote_new:nn`. This function is documented on page 8.)

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

```

457 \cs_new_protected:Npn \fnote_mark_gput:nn #1 #2 % #1 the representation of the mark, #2 type
458 {
459   \prop_gput:cxn { g__fnote_currentmarks_ #2 _prop }
460   { \tag_get:n{struct_num} }
461   { #1 }
462 }
463
464 \cs_generate_variant:Nn \fnote_mark_gput:nn {no,oo}

```

(End definition for `\fnote_mark_gput:nn`. This function is documented on page 8.)

`\fnote_mark_gpop:nnN` This commands takes as argument the representation of the mark (e.g. the content of `\@thefnmark`), the series (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

```

465 \cs_new_protected:Npn \fnote_mark_gpop:nnN #1 #2 #3
466 {
467   \seq_clear:N #3
468   \prop_set_eq:Nc \l__fnote_tmpa_prop { g__fnote_currentmarks_ #2 _prop }

```



```

469 \prop_map_inline:Nn \l__fnote_tmpa_prop
470 {
471   \tl_if_eq:nnT {#1} { ##2 }
472   {

```

store the key (the structure number) in the seq

```

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

```

remove entry as used from the global prop

```

474   \prop_gremove:cn { g__fnote_currentmarks_ #2 _prop } {##1}
475 }
476 }
477 }
478 \cs_generate_variant:Nn\fnote_mark_gpop:nnN {ooN}

```

(End definition for `\fnote_mark_gpop:nnN`. This function is documented on page 8.)

5.16.4 Enabling tagging and links for the mark command

To handle the mark in the text, we define a special command which will surround `\@makefnmark`. At this time `\@thefnmark` is already set.

```

479 \cs_new_protected:Npn \tag@FEMark #1 {%#1 content
480 {

```

end an open mc and start the structure.

```

481   \tag_mc_end_push:
482   \tag_struct_begin:n { tag=footnotemark }

```

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

```

483   \bool_if:NTF \l__fnote_autodetect_bool
484   {

```

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.

```

485   \fnote_mark_gput:oo {\@thefnmark}{\l__fnote_type_tl}
486   \tl_set:Nx \l__fnote_linktarget_tl {footnote*.\tag_get:n{struct_num}}
487 }

```

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.

```

488 {
489   \hook_gput_code:nne {tagpdf/finish/before} {tagpdf/footnote}
490   {
491     \exp_not:N\fnote_gput_refs:ee
492     { \tag_get:n{struct_num} }
493     { \ref_value:Vn\l__fnote_currentlabel_tl {fnote/struct} }
494   }

```

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

```

495   \tl_set:Nx \l__fnote_linktarget_tl {footnote*.\ref_value:Vn\l__fnote_currentlabel_tl}
496 }

```

And now the actual content

```

497 \tag_mc_begin:n{tag=Lbl}
498 %
499 \bool_if:NTF \l_fnote_link_bool
500 {
501     \exp_args:No
502     \hyper@linkstart
503     { \l_fnote_link_type_tl }
504     { \l__fnote_linktarget_tl }
505     #1
506     \hyper@linkend
507 }
508 { #1 }
509 \tag_mc_end:
510 \tag_struct_end:
511 \tag_mc_begin_pop:n{ }
512 }

```

At last set the kernel command to this one

```

513 \cs_set_eq:NN \@kernel@process@makefnmark \tag@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
514 \cs_new_protected:Npn \__fnote_gput_ref:nn #1 #2 % #1 the structure number receiving the ref
515 {
516     \tag_if_active:T
517     {
518         \tag_struct_gput:nne {#1}{ref}{\tag_struct_object_ref:e { #2 }}
519     }
520 }
521 \cs_new_protected:Npn \fnote_gput_refs:nn #1 #2 % pair of numbers
522 {
523     \__fnote_gput_ref:nn {#1}{#2}
524     \__fnote_gput_ref:nn {#2}{#1}
525 }
526 \cs_generate_variant:Nn \fnote_gput_refs:nn {ee}

```

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

```

527 \tl_new:N \l__fnote_dflt_struct_tl
528 \tl_set:Nn \l__fnote_dflt_struct_tl {1}

```

kernel hook for tagging (2 args) this sets the structure around the whole text

```

529 \cs_new_protected:Npn \tag@FENote #1#2 % #1 config point, \insert\footins #2 content
530 {
531     \tag_mc_end_push:

```

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

```

532 \tag_check_child:nnTF {FENote}{pdf2}
533 {
534     \tag_struct_begin:n { tag=footnote }
535 }

```

```

536     {
537       \tag_struct_begin:n { tag=footnote,parent=\l__fnote_dflt_struct_tl }
538     }

```

Store the current structure number for labels.

```

539     \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. Its number is `\l__fnote_currentstruct_tl` plus 1.

```

540     \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.

```

541     \bool_if:NTF \l__fnote_autodetect_bool
542     {

```

find open marks with identical `\@thefnmark`

```

543         \fnote_mark_gpop:ooN { \@thefnmark }{ \l_fnote_type_tl } \l__fnote_currentrefs_seq

```

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:

```

544         \seq_map_inline:Nn \l__fnote_currentrefs_seq
545         {
546             \fnote_gput_refs:ee {##1}{ \l__fnote_currentstruct_tl }
547         }
548     }

```

If no auto detection is done

```

549         {%no auto
550
551     }

```

This finishes 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`.

```

552         #1 {#2}
553     \tag_struct_end:
554     \tag_mc_begin_pop:n{ }
555 }

```

At last set the kernel command to this one

```

556 \cs_set_eq:NN \@kernel@process@footnotetext \tag@FENote

```

`\@makefntext@cfgpoint` is the configuration point responsible for typesetting the mark in the note. We use it to surround the mark with the needed tagging commands.

TODO check if this should/can be kernel configuration points or if an additional kernel configuration points are needed. If yes, what about the paragraph start and the paratagging??

```

557 \cs_set_protected:Npn \@makefntext@cfgpoint #1 %#1 code that typesets the mark.
558 {
559     \noindent
560     \tag_mc_end_push:
561     \tag@FENoteLbl { #1 }
562     \tag_mc_begin_pop:n{ }
563 }

```

`\tag@FENoteLbl` creates the label in the note on the bottom. It also adds link targets for the hyperlinking.

`\tag@FENoteLbl`

```
564 \cs_new_protected:Npn \tag@FENoteLbl #1
565 {
```

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)

```
566     %\seq_show:N\l__fnote_currentrefs_seq
567     \seq_map_inline:Nn\l__fnote_currentrefs_seq {\MakeLinkTarget*{footnote*.\##1}}
568     \MakeLinkTarget*{footnote*.\l__fnote_currentstruct_tl}
```

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

```
569     \tag_struct_begin:n { tag=NonStruct,parent=\l__fnote_currentstruct_tl +1 }
570     \tag_mc_begin:n { tag=Lbl }
571     #1
572     \tag_mc_end:
573     \tag_struct_end:
574 }
```

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

`\@makefntext@cfgpointii` is the configuration point around the actual note text.
 TODO check if this should/can be kernel configuration points

```
575 \cs_set_protected:Npn \@makefntext@cfgpointii #1
576 {
577     \tag_mc_end_push:
578     \tag@FENotetext { #1 }
579     \tag_mc_begin_pop:n{ }
580 }
```

`\tag@FENotetext`

This command currently only adds an MC chunk,
 TODO Should it set a mc or could it rely on the content?

```
581 \cs_new_protected:Npn \tag@FENotetext #1
582 {
583     \tag_mc_begin:n{ }
584     #1
585     \tag_mc_end:
586 }
```

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

```
587 \ExplSyntaxOff
588 </kernel>
589 <@@=)
```

6 Reimplementing the footmisc package

```

590 ⟨*footmisc⟩
591 %%
592 %% Copyright (c) 1995-2011 Robin Fairbairns
593 %% Copyright (c) 2018-2022 Robin Fairbairns, Frank Mittelbach
594 %%
595 %% This file is part of the ‘latex-lab Bundle’.
596 %% -----
597 %%
598 %% It may be distributed and/or modified under the
599 %% conditions of the LaTeX Project Public License, either version 1.3c
600 %% of this license or (at your option) any later version.
601 %% The latest version of this license is in
602 %%   https://www.latex-project.org/lppl.txt
603 %% and version 1.3c or later is part of all distributions of LaTeX
604 %% version 2008 or later.
605 %%
606 %% This work has the LPPL maintenance status ‘maintained’.
607 %%
608 %%
609 %% File: footmisc.dtx (C) Copyright 1995-2011 Robin Fairbairns
610 %%                  (C) Copyright 2018-2022 Frank Mittelbach
611 \NeedsTeXFormat{LaTeX2e}
612 \providecommand\DeclareRelease[3]{}
613 \providecommand\DeclareCurrentRelease[2]{}
614
615 \DeclareRelease{v5}{2011-06-06}{footmisc-2011-06-06.sty}
616 \DeclareCurrentRelease{}{2022-02-14}
617 \ProvidesPackage{latex-lab-footmisc}%
618   [2022/03/08 v6.0d
619     a miscellany of footnote facilities -- latex-lab version%
620   ]
621
622 \NeedsTeXFormat{LaTeX2e}[2020/10/01]
623 \newtoks\FN@temptoken
624 \providecommand\protected@writeaux{%
625   \protected@write\@auxout
626 }
627 \def\l@advance@macro{\@@d@vance@macro\edef}
628 \def\@@d@vance@macro#1#2#3{\expandafter\@tempcnta#2\relax
629   \advance\@tempcnta#3\relax
630   #1#2{\the\@tempcnta}%
631 }
632 \let\@advance@macro\l@advance@macro
633 \DeclareOption{symbol}{\renewcommand\thefootnote{\fnsymbol{footnote}}}
634 \newif\ifFN@robust \FN@robustfalse
635 \DeclareOption{symbol*}{%
636   \renewcommand\thefootnote{\@fnsymbol\c@footnote}%
637   \FN@robusttrue
638   \AtEndOfPackage{\setfnsymbol{lamport*-robust}}}%
639 }
640 \newif\ifFN@para \FN@parafalse
641 \DeclareOption{para}{%

```

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

```

642 % \PackageError{footmisc}{Option "\CurrentOption" incompatible with
643 % option "side"}%
644 % {I shall ignore "\CurrentOption"}%
645 \FN@paratrue
646 \setlength\footnotemargin{-\maxdimen} % default when para is used
647 }
648 \DeclareOption{side}{\ifFN@para
649 \PackageError{footmisc}{Option "\CurrentOption" incompatible with
650 option "para"}%
651 {I shall ignore "\CurrentOption"}%
652 \else
653 \def\@footnotetext@cfgpoint {\marginpar}
654 \AddToHook{fntext/para}{%
655 \hsize\marginparwidth % correct the default \hsize
656 \footnotesep\z@ % don't add a default separation
657 }
658 \def\@footnotetext@cfgpointii {\@makefntext}
659 \def\@footnotetext@cfgpointiii {}
660 \def\@footnotetext@cfgpointiv {}
661 \fi
662 }
663 \let\footnotelayout\@empty
664 \DeclareOption{ragged}{%
665 \@ifundefined{RaggedRight}%
666 {\renewcommand\footnotelayout{\linepenalty50 \raggedright}}%
667 {\renewcommand\footnotelayout{\linepenalty50 \RaggedRight}}%
668 }
669 \newif\ifFN@perpage
670 \FN@perpagefalse
671 \DeclareOption{perpage}{%
672 \FN@perpagetrue
673 }
674 \newif\ifFN@fixskip \FN@fixskipfalse
675
676 \let\FN@bottomcases\thr@@
677 \newif\ifFN@abovefloats \FN@abovefloatstrue
678 \DeclareOption{bottom}{%
679 \let\FN@bottomcases\@ne
680 \FN@abovefloatsfalse
681 \FN@fixskiptrue
682 }
683 \DeclareOption{bottomfloats}{%
684 \let\FN@bottomcases\tw@
685 \FN@abovefloatstrue \FN@fixskiptrue
686 }
687 \DeclareOption{abovefloats}{\FN@abovefloatstrue \FN@fixskiptrue}
688 \DeclareOption{belowfloats}{\FN@abovefloatsfalse \FN@fixskiptrue}
689 \DeclareOption{marginal}{%
690 \footnotemargin-0.8em\relax
691 }
692 \DeclareOption{flushmargin}{%
693 \footnotemargin0pt\relax

```

```

694 }
695 \newif\ifFN@hangfoot \FN@hangfootfalse
696 \DeclareOption{hang}{%
697   \FN@hangfoottrue
698 }
699 \newcommand*\hangfootparskip{0.5\baselineskip}
700 \newcommand*\hangfootparindent{0em}%
701 \DeclareOption{norule}{%
702   \renewcommand\footnoterule{%
703     \advance\skip\footins 4\p@\@plus2\p@\relax
704   }
705   \DeclareOption{splitrule}{%
706     \gdef\split@prev{0}
707     \let\pagefootnoterule\footnoterule
708     \let\mpfootnoterule\footnoterule
709     \def\splitfootnoterule{\kern-3\p@ \hrule \kern2.6\p@}
710     \def\footnoterule{\relax
711       \ifx \@listdepth\@mplistdepth
712         \mpfootnoterule
713       \else
714         \ifnum\split@prev=\z@
715           \pagefootnoterule
716         \else
717           \splitfootnoterule
718         \fi
719       \xdef\split@prev{\the\insertpenalties}%
720     }
721   }%
722 }
723 \newif\ifFN@stablefootnote \FN@stablefootnotefalse
724 \DeclareOption{stable}{\FN@stablefootnotetrue}
725 \newif\ifFN@multiplefootnote \FN@multiplefootnotefalse
726 \DeclareOption{multiple}{\FN@multiplefootnotetrue}
727 \ProcessOptions

```

This version of footmisc can assume that the new OR code is already available, thus nothing needs loading at this point. However, as long as we use this code also in a package version that can be loaded by other package while we are in a transition phase it is not clear whether not the kernel code is already available for other packages.

```

728 %\@ifundefined{kernel@before@cc1v}
729 % {\input{latex-lab-new-or.ltx}}{}

```

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

```

730 \ifFN@para
731   \def\@makecol@cfgpointii {%
732     \global\setbox\footins\vbox{\FN@makefootnoteparagraph}%
733   }
734 \fi
735 \ifFN@fixskip
736   \def\@outputbox@removebskip{%
737     \ifx\@textbottom\relax \else
738       \@outputbox@append{%
739         \@tempskipa\lastskip
740         \ifnum \gluestretchorder\@tempskipa>\z@

```

```

741         \vskip-\@tempskipa
742         \xdef\@outputbox@reinsertbskip
743             {\noexpand\@outputbox@append{\vskip\the\@tempskipa}}}%
744     \else
745         \global\let\@outputbox@reinsertbskip\relax
746     \fi
747 }%
748 \fi
749 }
750 \let\@outputbox@reinsertbskip\relax
751 \else
752     \let\@outputbox@removebskip \relax
753     \let\@outputbox@reinsertbskip\relax
754 \fi

755 \ifcase \FN@bottomcases\relax
756 \ERROR
757 \or
758     \ifFN@abovefloats
759         \def\@makecol@cfgpoint {%
760             \@if@footnotes@TF
761                 {\@outputbox@append{\vfill}}}%
762                 {\@if@bfloats@TF{\@outputbox@append{\vfill}}}%
763                 {\@outputbox@reinsertbskip}}%
764         \@outputbox@appendfootnotes
765         \@outputbox@attachfloats
766     }
767 \else
768     \def\@makecol@cfgpoint {%
769         \@outputbox@attachfloats
770         \@if@footnotes@TF
771             {\@outputbox@append{\vfill}}}%
772             {\@outputbox@reinsertbskip}%
773         \@outputbox@appendfootnotes
774     }
775 \fi
776 \or
777     \ifFN@abovefloats
778         \def\@makecol@cfgpoint {%
779             \@outputbox@appendfootnotes
780             \@if@bfloats@TF
781                 {\@outputbox@append{\vfill}}}%
782                 {\@outputbox@reinsertbskip}%
783             \@outputbox@attachfloats
784         }
785 \else
786     \def\@makecol@cfgpoint {%
787         \@if@footnotes@TF
788             {\@outputbox@append{\vfill}}}%
789             {\@if@bfloats@TF{\@outputbox@append{\vfill}}}%
790             {\@outputbox@reinsertbskip}}%
791         \@outputbox@attachfloats
792         \@outputbox@appendfootnotes
793     }
794 \fi

```



```

795 \or
796 \ifFN@abovefloats
797   \def\@makecol@cfgpoint {%
798     \@outputbox@appendfootnotes
799     \@outputbox@attachfloats
800     \@outputbox@reinsertbskip
801   }
802 \else
803   \def\@makecol@cfgpoint {%
804     \@outputbox@attachfloats
805     \@outputbox@appendfootnotes
806     \@outputbox@reinsertbskip
807   }
808 \fi
809 \else
810 \ERROR
811 \fi
812
813 % next can be dropped when cleaned up
814 \newif\ifFN@setspace
815 \ifpackageloaded{setspace}%
816 {%
817   \FN@setspace>true
818   \@ifclassloaded{memoir}%
819   {%
820     \AddToHook{fntext}{\let\baselinestretch\m@m@singlespace}%
821     \let\FN@baselinestretch\m@m@singlespace
822   }%
823   {%
824     \AddToHook{fntext}{\let\baselinestretch\setspace@singlespace}%
825     \let\FN@baselinestretch\setspace@singlespace
826   }%
827 }%
828 {%
829   \FN@setspace=false
830 }
831
832
833
834 \ifFN@para
835   \def\@footnotetext@cfgpoint {\insert\footins}
836
837   \long\def\@footnotetext@cfgpointii #1{%
838     \setbox\FN@tempboxa\hbox{\@makefntext{#1}}%
839     \dp\FN@tempboxa\z@
840     \ht\FN@tempboxa
841     \dimexpr\wd\FN@tempboxa *%
842       \footnotebaselineskip /\columnwidth\relax
843     \box\FN@tempboxa
844   }
845
846
847 \def\@footnotetext@cfgpointiii {}
848 \def\@footnotetext@cfgpointiv {% config point

```

```

849         \strut
850         \penalty-10\relax
851         \hskip\footglue
852     }
853 \fi
854
855
856
857 \ifFN@para
858     \let\FN@tempboxa\@tempboxa
859     \newbox\FN@tempboxb
860     \newbox\FN@tempboxc
861     \newskip\footglue \footglue=1em plus.3em minus.3em
862
863     %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
864     \newdimen\footnotebaselineskip
865
866     % establish late:
867
868     \AddToHook{begindocument/before} {%
869         {%
870             \footnotesize
871             \global\footnotebaselineskip=\normalbaselineskip
872         }%
873     }

```

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.

```

874
875     \long\def\FN@makefootnoteparagraph{%
876         \FN@setfootnoteparawidth
877         \@parboxrestore
878         \baselineskip=\footnotebaselineskip
879         \unvbox\footins \FN@removehboxes
880         \RawParEnd
881     }
882     \def\FN@removehboxes{\setbox\FN@tempboxa\lastbox
883         \ifhbox\FN@tempboxa{\FN@removehboxes}%
884         \unhbox\FN@tempboxa
885         \else
886             \RawNoindent
887             \rule{z@}{\footnotesep}
888         \fi
889     }
890 \fi
891
892
893 \@ifpackageloaded{multicol}
894 {
895     \def\FN@setfootnoteparawidth
896     {\hsize\ifnum\doublecol@number>\@ne
897         \textwidth
898         \else \columnwidth \fi}}
899 {\def\FN@setfootnoteparawidth{\hsize\columnwidth}}
900 \ifFN@perpage

```

```

901 \RequirePackage{perpage}
902 \MakePerPage{footnote}

```

Fix a bug in perpage ...

```

903 \def\@stpelt#1{\global\csname c@#1\endcsname \m@ne
904   \stepcounter{#1}%
905   \pp@fix@MakePerPage{#1}%
906 }
907 \def\pp@fix@MakePerPage#1{%
908   \ifnum \value{#1}>\z@
909     \addtocounter{#1}\m@ne\fi
910 }

```

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:

```

911 \def\pp@c1@end@iii\stepcounter#1\pp@fix@MakePerPage#2{}
912 \fi
913
914
915 \ifFN@para
916
917 % This can use the default interface, except that a negative value for
918 % \footnotemargin makes little sense, so we test for this and warn if
919 % necessary. But -\maxdimen is ok again, so would need to be a litte bit more elaborate.
920 %
921
922 %\AddToHook{fntext/para}{
923 % \ifdim \footnotemargin >\z@ \else
924 % \PackageWarningNoline{footmisc}{Option 'para' needs positive \noexpand\footnotemargin}%
925 % \footnotemargin 1.8em\relax
926 % \fi
927 %}
928
929
930 \AddToHook{fntext/begin}{\nobreak \hspace{.2em}}
931
932
933
934
935 \else
936
937 \ifFN@hangfoot
938 \long\def\@makefntext#1{%
939   \bgroup
940   \setbox\@tempboxa\hbox{%
941     \ifdim\footnotemargin>\z@
942       \hb@xt@\footnotemargin{\@makefnmark\hss}%
943     \else
944       \@makefnmark
945     \fi

```

```

946      }%
947      \leftmargin\wd\@tempboxa
948      \rightmargin\z@
949      \linewidth \columnwidth
950      \advance \linewidth -\leftmargin
951      \parshape \@ne \leftmargin \linewidth
952      \footnotesize
953      \@setpar{\@@par}}%
954      \leavevmode
955      \llap{\box\@tempboxa}%
956      \parskip\hangfootparskip\relax
957      \parindent\hangfootparindent\relax
958      \footnotelayout#1%
959      \par
960    \egroup
961  }
962
963  \else
964
965  % This is now using the default interface:
966  %
967  % \long\def\@makefntext#1{%
968  %   \parindent1em
969  %   \noindent
970  %   \ifdim\footnotemargin>\z@
971  %     \hb@xt@ \footnotemargin{\hss\@makefnmark}%
972  %   \else
973  %     \ifdim\footnotemargin=\z@
974  %       \llap{\@makefnmark}%
975  %     \else
976  %       \llap{\hb@xt@ -\footnotemargin{\@makefnmark\hss}}%
977  %     \fi
978  %   \fi
979  %   \footnotelayout#1%
980  % }
981
982  \fi
983  \fi
984
985
986
987
988  \ifFN@multiplefootnote
989    \providecommand*\multiplefootnotemarker}{3sp}
990  % we tag the separator as artifact
991  \ExplSyntaxOn
992  \providecommand*\multfootsep}{\tag_mc_end_push:\tag_mc_begin:n{artifact},\tag_mc_end:\tag
993  \ExplSyntaxOff
994  \AddToHook{fnmark}      {\FN@mf@check}
995  \AddToHook{fnmark/end} {\FN@mf@prepare}
996  %
997  \def\FN@mf@prepare{%
998    \kern-\multiplefootnotemarker
999    \kern\multiplefootnotemarker\relax

```

```

1000 }
1001 \def\FN@mfccheck{%
1002   \ifdim\lastkern=\multiplefootnotemarker\relax
1003   ??? is that necessary or even correct ??
1004   \edef\@x@sf{\the\spacefactor}%
1005   ??? shouldn't that be 2 unkerns ?? (none would also be ok)
1006   \unkern % new
1007   \unkern
1008   \textsuperscript{\multfootsep}%
1009   \spacefactor\@x@sf\relax
1010 \fi
1011 }
1012 \else
1013 \let\FN@mfcprepare\relax
1014 \fi
1015 \ifFN@stablefootnote
1016 \let\FN@sf@@footnote\footnote
1017 \def\footnote{\ifx\protect\@typeset@protect
1018   \expandafter\FN@sf@@footnote
1019   \else
1020     \expandafter\FN@sf@gobble@opt
1021   \fi
1022 }
1023 \edef\FN@sf@gobble@opt{\noexpand\protect
1024   \expandafter\noexpand\csname FN@sf@gobble@opt \endcsname}
1025 \expandafter\def\csname FN@sf@gobble@opt \endcsname{%
1026   \@ifnextchar[%]
1027     \FN@sf@gobble@twobrace
1028     \@gobble
1029 }
1030 \def\FN@sf@gobble@twobrace[#1]#2{}
1031 \let\FN@sf@@footnotemark\footnotemark
1032 \def\footnotemark{\ifx\protect\@typeset@protect
1033   \expandafter\FN@sf@@footnotemark
1034   \else
1035     \expandafter\FN@sf@gobble@optonly
1036   \fi
1037 }
1038 \edef\FN@sf@gobble@optonly{\noexpand\protect
1039   \expandafter\noexpand\csname FN@sf@gobble@optonly \endcsname}
1040 \expandafter\def\csname FN@sf@gobble@optonly \endcsname{%
1041   \@ifnextchar[%]
1042     \FN@sf@gobble@brace
1043     {}%
1044 }
1045 \def\FN@sf@gobble@brace[#1]{}
1046 \fi
1047 \newcommand\setfnsymbol[1]{%
1048   \@bsphack
1049   \@ifundefined{FN@fnsymbol@#1}%
1050   {%
1051     \PackageError{footmisc}{Symbol style "#1" not known}%
1052     \@eha
1053   }{%

```

```

1054     \expandafter\let\expandafter\@fnsymbol\csname
1055         FN@fnsymbol@#1\endcsname
1056     }%
1057     \@esphack
1058 }
1059 \let\FN@fnsymbol@lamport\@fnsymbol
1060 \newif\if@tempb
1061 \DeclareDocumentCommand\DefineFNSymbols {sm0{text}m}{%
1062     \expandafter\ifx\csname FN@fnsymbol@#2\endcsname\relax
1063         \PackageInfo{footmisc}{Declaring symbol style #2}%
1064     \else
1065         \PackageWarning{footmisc}{Redeclaring symbol style #2}%
1066     \fi
1067     \toks@{}%
1068     \def\@tempb{\end}%
1069     \FN@build@symboldef#4\end
1070     \def\@tempc{math}%
1071     \def\@tempd{#3}%
1072     \expandafter\xdef\csname FN@fnsymbol@#2\endcsname##1{%
1073         \ifx\@tempc\@tempd
1074             \noexpand\ensuremath
1075         \else
1076             \noexpand\nfss@text
1077         \fi
1078         {%
1079             \noexpand\ifcase##1%
1080             \the\toks@
1081             \noexpand\else
1082             \IfBooleanTF#1{\noexpand\@ctrerr}%
1083             {\noexpand\FN@orange##1}%
1084             \noexpand\fi
1085         }%
1086     }%
1087 }
1088 \def\FN@build@symboldef#1{%
1089     \def\@tempa{#1}%
1090     \ifx\@tempa\@tempb
1091     \else
1092         \toks@\expandafter{\the\toks@or#1}%
1093         \expandafter\FN@build@symboldef
1094     \fi
1095 }
1096 \DeclareDocumentCommand\DefineFNSymbolsTM {smm}{%
1097     \expandafter\ifx\csname FN@fnsymbol@#2\endcsname\relax
1098         \PackageInfo{footmisc}{Declaring symbol style #2}%
1099     \else
1100         \PackageWarning{footmisc}{Redeclaring symbol style #2}%
1101     \fi
1102     \toks@{}%
1103     \def\@tempb{\end}%
1104     \FN@build@symboldefTM#3\end\@null
1105     \expandafter\xdef\csname FN@fnsymbol@#2\endcsname##1{%
1106         \noexpand\ifcase##1%
1107         \the\toks@

```

```

1108     \noexpand\else
1109     \IfBooleanTF#1{\noexpand\@ctrerr}%
1110     {\noexpand\FN@orange##1}%
1111     \noexpand\fi
1112 }%
1113 }
1114 \def\FN@build@symboldefTM#1#2{%
1115   \def\@tempa{#1}%
1116   \ifx\@tempa\@tempb
1117   \else
1118     \toks@\expandafter{\the\toks@\or\TextOrMath{#1}{#2}}%
1119     \expandafter\FN@build@symboldefTM
1120   \fi
1121 }
1122 \def\FN@orange#1{%
1123   \ifFN@robust
1124     \@arabic#1%
1125     \bsphack
1126     \PackageInfo{footmisc}{Footnote number \number#1 out of range}%
1127     \protect\@fnsymbol@orange
1128     \esphack
1129   \else \@ctrerr \fi
1130 }
1131 \global\let\@diagnose@fnsymbol@orange\relax
1132 \AtEndDocument{\@diagnose@fnsymbol@orange}
1133 \def\@fnsymbol@orange{%
1134   \gdef\@diagnose@fnsymbol@orange{%
1135     \PackageWarningNoLine{footmisc}{Some footnote number(s)
1136       were out of range
1137       \MessageBreak
1138       see log for details%
1139     }%
1140   }%
1141 }
1142 \DefineFNSymbolsTM{bringhurst}{%
1143   \textasteriskcentered *%
1144   \textdagger \dagger
1145   \textdaggerdbl \ddagger
1146   \textsection \mathsection
1147   \textbardbl \||%
1148   \textparagraph \mathparagraph
1149 }%
1150 \DefineFNSymbolsTM{chicago}{%
1151   \textasteriskcentered *%
1152   \textdagger \dagger
1153   \textdaggerdbl \ddagger
1154   \textsection \mathsection
1155   \textbardbl \||%
1156   \#\#%
1157 }%
1158 \DefineFNSymbolsTM{wiley}{%
1159   \textasteriskcentered *%
1160   {\textasteriskcentered\textasteriskcentered}{**}%
1161   \textdagger \dagger

```

```

1162 \textdaggerdbl \ddagger
1163 \textsection \mathsection
1164 \textparagraph \mathparagraph
1165 \textbardbl \||
1166 }%
1167 \DefineFNsymbolsTM{lampport-robust}{%
1168 \textasteriskcentered *%
1169 \textdagger \dagger
1170 \textdaggerdbl \ddagger
1171 \textsection \mathsection
1172 \textparagraph \mathparagraph
1173 \textbardbl \||
1174 {\textasteriskcentered\textasteriskcentered}{**}%
1175 {\textdagger\textdagger}{\dagger\dagger}%
1176 {\textdaggerdbl\textdaggerdbl}{\ddagger\ddagger}%
1177 }
1178 \DefineFNsymbolsTM{lampport*}{%
1179 \textasteriskcentered *%
1180 \textdagger \dagger
1181 \textdaggerdbl \ddagger
1182 \textsection \mathsection
1183 \textparagraph \mathparagraph
1184 \textbardbl \||
1185 {\textasteriskcentered\textasteriskcentered}{**}%
1186 {\textdagger\textdagger}{\dagger\dagger}%
1187 {\textdaggerdbl\textdaggerdbl}{\ddagger\ddagger}%
1188 {\textsection\textsection}{\mathsection\mathsection}%
1189 {\textparagraph\textparagraph}{\mathparagraph\mathparagraph}%
1190 {\textasteriskcentered\textasteriskcentered\textasteriskcentered}{***}%
1191 {\textdagger\textdagger\textdagger}{\dagger\dagger\dagger}%
1192 {\textdaggerdbl\textdaggerdbl\textdaggerdbl}{\ddagger\ddagger\ddagger}%
1193 {\textsection\textsection\textsection}%%
1194 {\mathsection\mathsection\mathsection}%
1195 {\textparagraph\textparagraph\textparagraph}%%
1196 {\mathparagraph\mathparagraph\mathparagraph}%
1197 }
1198 \setfnsymbol{lampport*}
1199 \DefineFNsymbolsTM{lampport*-robust}{%
1200 \textasteriskcentered *%
1201 \textdagger \dagger
1202 \textdaggerdbl \ddagger
1203 \textsection \mathsection
1204 \textparagraph \mathparagraph
1205 \textbardbl \||
1206 {\textasteriskcentered\textasteriskcentered}{**}%
1207 {\textdagger\textdagger}{\dagger\dagger}%
1208 {\textdaggerdbl\textdaggerdbl}{\ddagger\ddagger}%
1209 {\textsection\textsection}{\mathsection\mathsection}%
1210 {\textparagraph\textparagraph}{\mathparagraph\mathparagraph}%
1211 {\textasteriskcentered\textasteriskcentered\textasteriskcentered}{***}%
1212 {\textdagger\textdagger\textdagger}{\dagger\dagger\dagger}%
1213 {\textdaggerdbl\textdaggerdbl\textdaggerdbl}{\ddagger\ddagger\ddagger}%
1214 {\textsection\textsection\textsection}%%
1215 {\mathsection\mathsection\mathsection}%

```



```

1216 {\textparagraph\textparagraph\textparagraph}%
1217 {\mathparagraph\mathparagraph\mathparagraph}%
1218 }
1219 \newcommand\mpfootnotemark{%
1220 \@ifnextchar[%
1221 \@xmpfootnotemark
1222 {%
1223 \stepcounter\@mpfn
1224 \protected@xdef\@thefnmark{\thempfn}%
1225 \@footnotemark
1226 }%
1227 }
1228 \def\@xmpfootnotemark[#1]{%
1229 \begingroup
1230 \csname c@\@mpfn\endcsname #1\relax
1231 \unrestored@protected@xdef\@thefnmark{\thempfn}%
1232 \endgroup
1233 \@footnotemark
1234 }
TEMP PATCHES FOR TESTING
1235
1236 \endinput
1237 </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			
\#	1156	\bool_new:N	68, 75, 118
\	1147, 1155, 1165, 1173, 1184, 1205	\bool_set_false:N	388, 428, 431
		\bool_set_true:N	69, 76, 391
		\box	843, 955
A			
\addtocounter	909	C	
\AddToHook	276, 346, 407, 409, 415, 417, 436, 444, 654, 820, 824, 868, 922, 930, 994, 995	\columnwidth	15, 224, 842, 897, 898, 949
\advance	629, 703, 950	\cs	232, 233
\AtBeginDocument	283, 419, 424	cs commands:	
\AtEndDocument	1132	\cs_generate_variant:Nn	79, 80, 81, 82, 83, 464, 478, 526
\AtEndOfPackage	638	\cs_gset_eq:NN	129, 148
B		\cs_if_eq:NNTF	348
\baselineskip	699, 878	\cs_log:N	136, 137, 138, 139, 140, 141
\baselinestretch	417, 820, 824	\cs_new:Npn	338, 340, 342, 344
\begingroup	385, 1229	\cs_new_eq:NN	195, 316
\bgroup	939	\cs_new_protected:Npn	84, 97, 120, 132, 151, 179, 198, 271, 272, 273, 274, 275, 289, 309, 310, 318, 393, 435, 451, 457, 465, 479, 514, 521, 529, 564, 581
bool commands:		\cs_set:Npn	335
\bool_gset_true:N	119		
\bool_if:NNTF	122, 134, 483, 499, 541		

<code>\cs_set_eq:NN</code>	<code>\exp_not:N</code>	491
.... 197, 278, 322, 325, 328, 330,	<code>\exp_stop_f:</code>	330, 339, 341, 343
350, 372, 374, 411, 412, 413, 513, 556	<code>\expandafter</code>	
<code>\cs_set_protected:Npn</code> . 446, 557, 575	628, 1018, 1020, 1024, 1025, 1033,	
<code>\csname</code> 19, 903, 1024, 1025, 1039, 1040,	1035, 1039, 1040, 1054, 1062, 1072,	
1054, 1062, 1072, 1097, 1105, 1230	1092, 1093, 1097, 1105, 1118, 1119	
<code>\CurrentOption</code>	<code>\ExplSyntaxOff</code>	401, 587, 993
642, 644, 649, 651	<code>\ExplSyntaxOn</code>	63, 406, 991
D		
<code>\dagger</code> .. 1144, 1152, 1161, 1169, 1175,	F	
1180, 1186, 1191, 1201, 1207, 1212	<code>\fi</code>	169, 188,
<code>\ddagger</code> . 1145, 1153, 1162, 1170, 1176,	287, 302, 303, 304, 661, 718, 720,	
1181, 1187, 1192, 1202, 1208, 1213	734, 746, 748, 754, 775, 794, 808,	
<code>\DeclareCurrentRelease</code>	811, 853, 888, 890, 897, 909, 912,	
613, 616	926, 945, 977, 978, 982, 983, 1010,	
<code>\DeclareDocumentCommand</code>	1014, 1021, 1036, 1046, 1066, 1077,	
..... 364, 369, 378, 383, 1061, 1096	1084, 1094, 1101, 1111, 1120, 1129	
<code>\DeclareOption</code>	<code>\floatingpenalty</code>	223
633, 635,	fnote commands:	
641, 648, 664, 671, 678, 683, 687,	<code>\g_fnote_debug_bool</code> 118, 119, 122, 134	
688, 689, 692, 696, 701, 705, 724, 726	<code>\fnote_footnotemark:</code> <u>151</u> , 151, 197, 411	
<code>\DeclareRelease</code>	<code>\fnote_footnotetext:n</code>	
612, 615 <u>198</u> , 198, 278, 416	
<code>\def</code>	<code>\fnote_gput_refs:nn</code>	
11, 17, 28, 491, <u>514</u> , 521, 526, 546	
29, 33, 196, 227, 335, 418, 627, 628,	<code>\l_fnote_link_bool</code> <u>23</u> , <u>68</u> , 428, 431, 499	
653, 658, 659, 660, 709, 710, 731,	<code>\l_fnote_link_type_tl</code>	<u>70</u> , 503
736, 759, 768, 778, 786, 797, 803,	<code>\fnote_makefn-text:n</code>	
835, 837, 847, 848, 875, 882, 894, <u>9</u> , <u>19</u> , <u>20</u> , <u>289</u> , 289, 350	
898, 903, 907, 911, 938, 967, 997,	<code>\fnote_mark_gpop:nnN</code>	
1001, 1017, 1025, 1030, 1032, 1040, <u>8</u> , <u>465</u> , 465, 478, 543	
1045, 1068, 1070, 1071, 1088, 1089,	<code>\fnote_mark_gput:nn</code>	
1103, 1114, 1115, 1122, 1133, 1228 <u>8</u> , <u>457</u> , 457, 464, 485	
<code>\DefineFNsymbols</code>	<code>\fnote_new:nn</code>	<u>8</u> , <u>451</u> , 451, 456
1061	<code>\fnote_set_fnmark:nn</code> <u>97</u> , 97, 366	
<code>\DefineFNsymbolsTM</code>	<code>\fnote_step_fnmark:nn</code> <u>84</u> , 84, 371, 380	
1096,	<code>\l_fnote_type_tl</code>	<u>66</u> , 485, 543
1142, 1150, 1158, 1167, 1178, 1199	fnote internal commands:	
<code>\dimexpr</code>	<code>\l__fnote_autodetect_bool</code>	
841 <u>75</u> , 388, 391, 483, 541	
<code>\dp</code>	<code>\l__fnote_currentlabel_tl</code>	
222, 839 <u>21</u> , <u>25</u> , <u>73</u> , 389, 493, 495	
E		
<code>\edef</code>	<code>\l__fnote_currentrefs_seq</code>	
160, 627, 1004, 1023, 1038 <u>74</u> , 543, 544, 566, 567	
<code>\egroup</code>	<code>\l__fnote_currentstruct_tl</code>	
960	... <u>23</u> , <u>77</u> , <u>78</u> , 434, 539, 546, 568, 569	
<code>\else</code> 294, 297, 300, 652, 713, 716,	<code>__fnote_debug_footnotemark:</code> ...	
737, 744, 751, 767, 785, 802, 809, 120, 129, 152	
885, 897, 923, 935, 943, 963, 972,	<code>__fnote_debug_footnotetext:</code> ...	
975, 1012, 1019, 1034, 1064, 1075, 132, 148, 199	
1081, 1091, 1099, 1108, 1117, 1129	<code>\l__fnote_dflt_struct_tl</code> 527, 528, 537	
<code>\end</code>	<code>__fnote_finish:</code> ... <u>179</u> , 179, 195, 374	
1068, 1069, 1103, 1104	<code>__fnote_footmark_finish:</code>	15
<code>\endcsname</code>		
... 19, 903, 1024, 1025, 1039, 1040,		
1055, 1062, 1072, 1097, 1105, 1230		
<code>\endgroup</code>		
387, 1232		
<code>\endinput</code>		
1236		
<code>\ensuremath</code>		
1074		
<code>\ERROR</code> <u>20</u> , 316, 329, 362, 405, 756, 810		
exp commands:		
<code>\exp_after:wN</code>		334, 336
<code>\exp_args:No</code>		501

<code>__fnote_footnotemark_finish:</code> . . .	<code>\ht</code> 840
. 177, 179, 195, 372, 374, 376	
<code>__fnote_gput_ref:nn</code> 514, 514, 523, 524	
<code>__fnote_label_hook:</code> 435, 435, 441, 446	
<code>\l_fnote_linktarget_tl</code>	
. 72, 486, 495, 504	
<code>__fnote_patch:</code> 318, 353	
<code>__fnote_patch_@makefnmark:w</code> 328, 342	
<code>__fnote_patch_hb@xt@:w</code> 325, 340	
<code>__fnote_patch_hbox:w</code> 322, 338	
<code>\l_fnote_patch_seq</code> 361, 362	
<code>\l__fnote_patch_tl</code> 315,	
320, 321, 324, 327, 333, 334, 336, 361	
<code>__fnote_tmp:w</code>	
316, 322, 325, 328, 330, 334, 335, 336	
<code>\l_fnote_tmpa_prop</code> 64, 468, 469	
<code>\l_fnote_tmpa_tl</code> 65	
<code>\fnsymbol</code> 633	
<code>\footglue</code> 851, 861	
<code>\footins</code> . . . 4, 271, 529, 703, 732, 835, 879	
<code>\footnote</code> 7, 9, 369, 1016, 1017	
<code>\footnotebaselineskip</code> . 842, 864, 871, 878	
<code>\footnotelayout</code> . . 663, 666, 667, 958, 979	
<code>\footnotemargin</code> 6, 18,	
280, 281, 285, 286, 292, 293, 295,	
298, 301, 646, 690, 693, 918, 923,	
924, 925, 941, 942, 970, 971, 973, 976	
<code>\footnotemark</code> . . 3, 4, 7-9, 378, 1031, 1032	
<code>\footnoterule</code> 702, 707, 708, 710	
<code>\footnotesep</code> 22, 221, 273, 656, 887	
<code>\footnotesize</code> 14, 214, 870, 952	
<code>\footnotetext</code> 5-9, 364	
<code>\footref</code> 8, 9, 21, 23, 383	
G	
<code>\gdef</code> 706, 1134	
<code>\global</code> . . 12, 395, 732, 745, 871, 903, 1131	
<code>\gluestretchorder</code> 740	
group commands:	
<code>\group_begin:</code> 91, 103	
<code>\group_end:</code> 94, 106	
H	
<code>\hangfootparindent</code> 700, 957	
<code>\hangfootparskip</code> 699, 956	
<code>\hbox</code> 19,	
20, 33, 321, 338, 339, 358, 361, 838, 940	
hook commands:	
<code>\hook_gput_code:nnn</code> 82, 489	
<code>\hrule</code> 709	
<code>\hsize</code> 5, 16, 15, 224, 655, 895, 898	
<code>\hskip</code> 851	
<code>\hspace</code> 930	
<code>\hss</code> 293, 301, 314, 942, 971, 976	
I	
<code>\IfBooleanTF</code> 1082, 1109	
<code>\ifcase</code> 755, 1079, 1106	
<code>\ifdim</code> 285,	
292, 295, 298, 923, 941, 970, 973, 1002	
<code>\ifhbox</code> 883	
<code>\ifhmode</code> 159, 186	
<code>\ifnum</code> 714, 740, 895, 908	
<code>\ifx</code> 711, 737, 1017,	
1032, 1062, 1073, 1090, 1097, 1116	
<code>\ignorespaces</code> 22, 254	
<code>\indent</code> 5	
<code>\input</code> 729	
<code>\insert</code> 4, 15, 271, 529, 835	
<code>\insertpenalties</code> 719	
int commands:	
<code>\int_compare:nTF</code> 362	
<code>\int_set:Nn</code> 92, 104	
<code>\interfootnotelinepenalty</code> 220	
<code>\interlinepenalty</code> 5, 220	
K	
<code>\kern</code> 709, 998, 999	
L	
<code>\label</code> 8, 9	
<code>\lastbox</code> 882	
<code>\lastkern</code> 1002	
<code>\lastskip</code> 739	
<code>\leavevmode</code> 158, 954	
<code>\leftmargin</code> 947, 950, 951	
legacy commands:	
<code>\legacy_if:nTF</code> 428	
<code>\let</code> 403, 404, 408, 416, 417, 632,	
663, 676, 679, 684, 707, 708, 745,	
750, 752, 753, 820, 821, 824, 825,	
858, 1013, 1016, 1031, 1054, 1059, 1131	
<code>\linepenalty</code> 666, 667	
<code>\linewidth</code> 949, 950, 951	
<code>\llap</code> 6, 296, 301, 955, 974, 976	
<code>\LogHook</code> 124, 125, 126,	
127, 128, 142, 143, 144, 145, 146, 147	
<code>\long</code> 11, 335, 837, 875, 938, 967	
M	
<code>\MakeLinkTarget</code> 567, 568	
<code>\MakePerPage</code> 35, 902	
<code>\marginpar</code> 653	
<code>\marginparwidth</code> 655	
<code>\mathparagraph</code> 1148, 1164, 1172,	
1183, 1189, 1196, 1204, 1210, 1217	
<code>\mathsection</code> . . 1146, 1154, 1163, 1171,	
1182, 1188, 1194, 1203, 1209, 1215	

<code>\maxdimen</code>	6, 281, 285, 298, 646, 919	<code>\providecommand</code>	
<code>\MessageBreak</code>	1137	421, 422, 612, 613, 624, 989, 992
<code>\mpfootnotemark</code>	1219	<code>\ProvidesFile</code>	3
<code>\mpfootnoterule</code>	708, 712	<code>\ProvidesPackage</code>	617
<code>\multfootsep</code>	992, 1008		
<code>\multiplefootnotemark</code>	989, 998, 999, 1002		
		R	
N		<code>\RaggedRight</code>	667
<code>\NeedsTeXFormat</code>	611, 622	<code>\raggedright</code>	666
<code>\newbox</code>	859, 860	<code>\RawNoindent</code>	886
<code>\newcommand</code>	311, 699, 700, 1047, 1219	<code>\RawParEnd</code>	880
<code>\newdimen</code>	280, 864	<code>\ref</code>	21, 386
<code>\NewHook</code>	110, 111, 112, 114, 115, 116, 117	ref commands:	
<code>\newif</code>	634, 640,	<code>\ref_attribute_gset:nnnn</code>	434
669, 674, 677, 695, 723, 725, 814, 1060		<code>\ref_label:nn</code>	79, 448
<code>\NewMirroredHookPair</code>	109, 113	<code>\ref_value:nn</code>	80, 493, 495
<code>\newskip</code>	861	<code>\relax</code>	187, 628, 629, 690, 693, 703,
<code>\newtoks</code>	623	710, 737, 745, 750, 752, 753, 755,	
<code>\nobreak</code>	4, 168, 930	842, 850, 925, 956, 957, 999, 1002,	
<code>\noexpand</code>		1009, 1013, 1062, 1097, 1131, 1230	
. 743, 924, 1023, 1024, 1038, 1039,		<code>\renewcommand</code>	633, 636, 666, 667, 702
1074, 1076, 1079, 1081, 1082, 1083,		<code>\RequirePackage</code>	400, 438, 901
1084, 1106, 1108, 1109, 1110, 1111		<code>\rightmargin</code>	948
<code>\noindent</code>	309, 313, 559, 969	<code>\rule</code>	22, 273, 887
<code>\normalbaselineskip</code>	871		
<code>\normalfont</code>	33	S	
<code>\number</code>	1126	seq commands:	
		<code>\seq_clear:N</code>	467
O		<code>\seq_count:N</code>	362
<code>\or</code>	757, 776, 795, 1092, 1118	<code>\seq_map_inline:Nn</code>	544, 567
		<code>\seq_new:N</code>	74
P		<code>\seq_put_right:Nn</code>	473
<code>\PackageError</code>	642, 649, 1051	<code>\seq_set_split:Nnn</code>	361
<code>\PackageInfo</code>	1063, 1098, 1126	<code>\seq_show:N</code>	566
<code>\PackageWarning</code>	1065, 1100	<code>\setbox</code>	12, 395, 732, 838, 882, 940
<code>\PackageWarningNoLine</code>	1135	<code>\setfnsymbol</code>	638, 1047, 1198
<code>\PackageWarningNoline</code>	924	<code>\setlength</code>	286, 646
<code>\pagefootnoterule</code>	707, 715	<code>\skip</code>	703
<code>\par</code>	23, 263, 959	<code>\spacefactor</code>	160, 187, 1004, 1009
<code>\parindent</code>	5, 226, 312, 957, 968	<code>\splitfootnoterule</code>	709, 717
<code>\parshape</code>	951	<code>\splitmaxdepth</code>	222
<code>\parskip</code>	956	<code>\splittopskip</code>	221
<code>\patchcmd</code>	402	<code>\stepcounter</code>	35, 87, 904, 911, 1223
<code>\penalty</code>	850	<code>\strut</code>	849
prg commands:		<code>\strutbox</code>	22, 222, 274
<code>\prg_do_nothing:</code>	129, 148, 372		
<code>\ProcessOptions</code>	727	T	
prop commands:		tag commands:	
<code>\prop_gput:Nnn</code>	81, 459	<code>\tag_check_child:nnTF</code>	532
<code>\prop_gremove:Nn</code>	474	<code>\tag_get:n</code>	460, 486, 492, 539
<code>\prop_map_inline:Nn</code>	469	<code>\tag_if_active:TF</code>	516
<code>\prop_new:N</code>	64, 453	<code>\tag_mc_begin:n</code>	497, 570, 583, 992
<code>\prop_set_eq:NN</code>	468	<code>\tag_mc_begin_pop:n</code>	
<code>\protect</code>	1017, 1023, 1032, 1038, 1127	511, 554, 562, 579, 992
		<code>\tag_mc_end:</code>	509, 572, 585, 992

<code>\tag_mc_end_push:</code>	481, 531, 560, 577, 992
<code>\tag_struct_begin:n</code>	482, 534, 537, 540, 569
<code>\tag_struct_end:</code> ...	510, 540, 553, 573
<code>\tag_struct_gput:nnn</code>	518
<code>\tag_struct_object_ref:n</code>	518
<code>\tag_struct_use:n</code>	83
TeX and L ^A T _E X 2 _ε commands:	
<code>\@@dvance@macro</code>	627, 628
<code>\@@par</code>	953
<code>\@MM</code>	223
<code>\@advance@macro</code>	632
<code>\@arabic</code>	1124
<code>\@auxout</code>	625
<code>\@bsphack</code>	1048, 1125
<code>\@ctrerr</code>	1082, 1109, 1129
<code>\@currentcounter</code>	17, 227
<code>\@currentlabel</code>	18, 228
<code>\@diagnose@fnsymbol@orange</code>	1131, 1132, 1134
<code>\@eha</code>	1052
<code>\@empty</code>	422, 663
<code>\@esphack</code>	1057, 1128
<code>\@finalstrut</code>	22, 274
<code>\@fnsymbol</code>	636, 1054, 1059
<code>\@fnsymbol@orange</code>	1127, 1133
<code>\@footnotemark</code>	3, 15, 21, 197, 373, 381, 390, 411, 1225, 1233
<code>\@footnotetext</code>	3, 16, 278, 367, 375, 403, 416
<code>\@footnotetext@cfgpoint</code>	4, 9, 136, 205, 271, 404, 653, 835
<code>\@footnotetext@cfgpointii</code>	4, 5, 137, 243, 272, 658, 837
<code>\@footnotetext@cfgpointiii</code>	4, 138, 248, 273, 659, 847
<code>\@footnotetext@cfgpointiv</code>	5, 139, 261, 274, 660, 848
<code>\@gobble</code>	1028
<code>\@gobbletwo</code>	421
<code>\@if@bfloats@TF</code>	762, 780, 789
<code>\@if@footnotes@TF</code>	760, 770, 787
<code>\@ifclassloaded</code>	818
<code>\@ifnextchar</code>	1026, 1041, 1220
<code>\@ifpackageloaded</code>	426, 815, 893
<code>\@ifundefined</code>	665, 728, 1049
<code>\@iiiminipage</code>	22, 402
<code>\@kernel@process@footnotetext</code>	9, 204, 275, 556
<code>\@kernel@process@makefnmark</code>	9, 174, 196, 513
<code>\@listdepth</code>	711
<code>\@makecol</code>	408, 413
<code>\@makecol@cfgpoint</code>	759, 768, 778, 786, 797, 803
<code>\@makecol@cfgpointii</code>	731
<code>\@makefnmark</code>	3, 4, 6, 9, 19, 33, 175, 293, 296, 299, 301, 314, 327, 342, 343, 942, 944, 971, 974, 976
<code>\@makefntext</code>	3–6, 9, 10, 18–20, 25, 21, 226, 240, 272, 320, 335, 348, 350, 412, 658, 838, 938, 967
<code>\@makefntext@cfgpoint</code>	5, 9, 10, 27, 140, 290, 309, 344, 557
<code>\@makefntext@cfgpointii</code>	5, 9, 10, 28, 141, 306, 310, 320, 575
<code>\@makefntext@processX</code>	339, 341, 343, 344
<code>\@mpfn</code>	28, 366, 371, 1223, 1230
<code>\@mpfootins</code>	12, 13, 395, 396
<code>\@mpfootnotetext</code>	8, 11, 403
<code>\@mpfootnotetext@cfgpoint</code> ..	393, 404
<code>\@mplistdepth</code>	711
<code>\@one</code>	679, 895, 951
<code>\@null</code>	1104
<code>\@outputbox@append</code>	738, 743, 761, 762, 771, 781, 788, 789
<code>\@outputbox@appendfootnotes</code>	764, 773, 779, 792, 798, 805
<code>\@outputbox@attachfloats</code>	765, 769, 783, 791, 799, 804
<code>\@outputbox@reinsertbskip</code>	750, 753, 763, 772, 782, 790, 800, 806
<code>\@outputbox@removebskip</code>	736, 752
<code>\@parboxrestore</code>	16, 225, 877
<code>\@plus</code>	703
<code>\@setpar</code>	953
<code>\@stpelt</code>	903
<code>\@tempa</code>	1089, 1090, 1115, 1116
<code>\@tempb</code>	1068, 1090, 1103, 1116
<code>\@tempboxa</code>	858, 940, 947, 955
<code>\@tempc</code>	1070, 1073
<code>\@tempcnta</code>	628, 629, 630
<code>\@tempd</code>	1071, 1073
<code>\@tempskipa</code>	739, 740, 741, 743
<code>\@textbottom</code>	737
<code>\@textsuperscript</code>	33
<code>\@thefnmark</code>	7–9, 11, 13, 21, 24, 25, 27, 19, 33, 88, 93, 100, 105, 228, 386, 485, 543, 1224, 1231
<code>\@typeset@protect</code>	1017, 1032
<code>\@x@sf</code>	160, 187, 1004, 1009
<code>\@xmpfootnotemark</code>	1221, 1228
<code>\@iiiminipage</code>	10
<code>\c@footnote</code>	636
<code>\color@begingroup</code>	20, 238
<code>\color@endgroup</code>	24, 264

<code>\doublecol@number</code>	895	<code>\ifFN@robust</code>	634, 1123
<code>\FN@abovefloatsfalse</code>	680, 688	<code>\ifFN@setspace</code>	814
<code>\FN@abovefloatstrue</code> ...	677, 685, 687	<code>\ifFN@stablefootnote</code>	723, 1015
<code>\FN@baselinestretch</code>	821, 825	<code>\l@advance@macro</code>	627, 632
<code>\FN@bottomcases</code> ...	676, 679, 684, 755	<code>\label@hook</code>	439
<code>\FN@build@symboldef</code> .	1069, 1088, 1093	<code>\label@name</code>	24, 448
<code>\FN@build@symboldefTM</code>	1104, 1114, 1119	<code>\m@m@singlespace</code>	820, 821
<code>\FN@fixskipfalse</code>	674	<code>\m@ne</code>	903, 909
<code>\FN@fixskiptrue</code> ...	681, 685, 687, 688	<code>\new@std@class@makecol</code>	408, 413
<code>\FN@fnsymbol@lamport</code>	1059	<code>\nfss@text</code>	1076
<code>\FN@hangfootfalse</code>	695	<code>\old@std@class@makefntext</code>	
<code>\FN@hangfoottrue</code>	697	311, 348, 412
<code>\FN@makefootnoteparagraph</code> ..	732, 875	<code>\p@</code>	703, 709
<code>\FN@mf@check</code>	994, 1001	<code>\p@footnote</code>	228
<code>\FN@mf@prepare</code>	995, 997, 1013	<code>\pp@cl@end@iii</code>	911
<code>\FN@multiplefootnotefalse</code>	725	<code>\pp@fix@MakePerPage</code> ...	905, 907, 911
<code>\FN@multiplefootnotetrue</code>	726	<code>\protected@edef</code>	18, 228
<code>\FN@orange</code>	1083, 1110, 1122	<code>\protected@write</code>	625
<code>\FN@parafalse</code>	640	<code>\protected@writeaux</code>	624
<code>\FN@paratrue</code>	645	<code>\protected@xdef</code>	88, 100, 1224
<code>\FN@perpagefalse</code>	670	<code>\reset@font</code>	14, 213
<code>\FN@perpagetrue</code>	672	<code>\setspace@singlespace</code> .	417, 824, 825
<code>\FN@removehboxes</code>	879, 882, 883	<code>\split@prev</code>	706, 714, 719
<code>\FN@robustfalse</code>	634	<code>\tag@FEMark</code>	9, 479, 513
<code>\FN@robusttrue</code>	637	<code>\tag@FENote</code>	529, 556
<code>\FN@setfootnoteparawidth</code>	876, 894, 898	<code>\tag@FENoteLbl</code>	5, 28, 561, 564
<code>\FN@setspacefalse</code>	829	<code>\tag@FENotetext</code>	5, 578, 581
<code>\FN@setspacetrue</code>	817	<code>\thr@@</code>	676
<code>\FN@sf@@footnote</code>	1016, 1018	<code>\toks@</code> 1067, 1080, 1092, 1102, 1107, 1118	
<code>\FN@sf@@footnotemark</code> ...	1031, 1033	<code>\tw@</code>	684
<code>\FN@sf@gobble@bracket</code> ...	1042, 1045	<code>\unrestored@protected@xdef</code>	
<code>\FN@sf@gobble@opt</code>	1020, 1023	93, 105, 386, 1231
<code>\FN@sf@gobble@optonly</code> ...	1035, 1038	<code>\z@</code> .	22, 273, 292, 295, 656, 714, 740,
<code>\FN@sf@gobble@twobracket</code> .	1027, 1030		839, 887, 908, 923, 941, 948, 970, 973
<code>\FN@stablefootnotefalse</code>	723	<code>\textasteriskcentered</code>	
<code>\FN@stablefootnotetrue</code>	724		1143, 1151, 1159, 1160, 1168, 1174,
<code>\FN@tempboxa</code>	838,		1179, 1185, 1190, 1200, 1206, 1211
	839, 840, 841, 843, 858, 882, 883, 884	<code>\textbardbl</code>	
<code>\FN@tempboxb</code>	859	..	1147, 1155, 1165, 1173, 1184, 1205
<code>\FN@tempboxc</code>	860	<code>\textdagger</code>	1144, 1152, 1161, 1169, 1175,
<code>\FN@temptoken</code>	623		1180, 1186, 1191, 1201, 1207, 1212
<code>\g@addto@macro</code>	439	<code>\textdaggerdbl</code>	
<code>\hb@xt@</code>	19, 293,		1145, 1153, 1162, 1170, 1176,
	301, 314, 324, 340, 341, 942, 971, 976		1181, 1187, 1192, 1202, 1208, 1213
<code>\hyper@linkend</code>	422, 506	<code>\TextOrMath</code>	1118
<code>\hyper@linkstart</code>	421, 502	<code>\textparagraph</code>	1148, 1164, 1172,
<code>\hyper@nopatch@footnote</code>	418		1183, 1189, 1195, 1204, 1210, 1216
<code>\if@tempswb</code>	1060	<code>\textsection</code> ..	1146, 1154, 1163, 1171,
<code>\ifFN@abovefloats</code> ..	677, 758, 777, 796		1182, 1188, 1193, 1203, 1209, 1214
<code>\ifFN@fixskip</code>	674, 735	<code>\textsuperscript</code>	1008
<code>\ifFN@hangfoot</code>	695, 937	<code>\textwidth</code>	896
<code>\ifFN@multiplefootnote</code>	725, 988	<code>\the</code>	160, 630,
<code>\ifFN@para</code> .	640, 648, 730, 834, 857, 915		719, 743, 1004, 1080, 1092, 1107, 1118
<code>\ifFN@perpage</code>	669, 900	<code>\thefootnote</code>	29, 633, 636

<code>\thempfn</code>	29, 1224, 1231	<code>\use:n</code>	343
tl commands:			
<code>\tl_if_eq:nnTF</code>	471	<code>\UseHook</code>	156, 166, 172, 184, 191, 202, 211, 236, 252, 259, 268
<code>\tl_if_in:NnTF</code>	321, 324, 327		
<code>\tl_if_novalue:nTF</code>	85, 98		
<code>\tl_new:N</code> 65, 66, 70, 72, 73, 77, 315, 527		V	
<code>\tl_set:Nn</code>	67, 71, 78, 320, 333, 389, 486, 495, 528, 539	<code>\value</code>	908
		<code>\vbox</code>	12, 395, 732
		<code>\vfill</code>	761, 762, 771, 781, 788, 789
		<code>\vskip</code>	741, 743
U			
<code>\unhbox</code>	884		
<code>\unkern</code>	1006, 1007	W	
<code>\unskip</code>	155	<code>\wd</code>	841, 947
<code>\unvbox</code>	13, 396, 879		
use commands:			
<code>\use:N</code>	88, 93, 100, 105	X	
		<code>\xdef</code>	719, 742, 1072, 1105