The `latex-lab-footnotes` code*

Frank Mittelbach, LaTeX Project

March 9, 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 `\footref` . . . 8
3.3.2 `\footnotemark` after `\footnotetext` . . . 8
3.4 Links . . . 9
3.5 Implementation details . . . 9
3.6 Handling the mark . . . 9
3.7 Handling the footnotetext . . . 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 `\@thefnmark` . . . 13
5.8 Hooks . . . 13
5.9 Debugging code . . . 14
5.10 The new `\@footnotemark` command . . . 15
5.11 The new `\@footnotetext` command . . . 16
5.12 The new `\@makefntext` command . . . 18
5.12.1 Making documents use the new `\@makefntext` . . . 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 LaTeX 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 LaTeX'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 mark[1] 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`.

[1] Like 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, scrlttr2, 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 `\@makefntext` which contains various hooks for customization. For most scenarios this is sufficient. However, when running all footnotes as a single paragraph at the bottom, then each footnote needs to be prepared prior to storing 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@FENoteLbl`.

`\@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 this[1], and a *footnotetext* (short: note) that is placed at the bottom of the page. The *footnotetext* normally repeats at the begin the mark as a visual clue.

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

The mark doesn't need to be visible, e.g. the typesetted mark[1–3] denotes three marks, where the second is invisible. Tagging should produce here probably three `Lbl` 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 `/Lbl` structure[2] 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 `/Lbl`[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

[2] to make it easier to identify the role we use `/footnotemark` which we rolemap to `/Lbl`

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

[4] We tag it as `/footnotelabel`.

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

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

The default property uses the name `default`

`\fnote_new:nn` `\fnote_new:nn{⟨name⟩}{⟨key/value option⟩}`

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.

`\fnote_mark_gput:nn` `\fnote_mark_gput:nn{⟨mark⟩}{⟨footnote type name⟩}`
`\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*⟩

`\fnote_mark_gpop:nnN` `\fnote_mark_gpop:nnN{⟨mark⟩}{⟨footnote type name⟩}⟨sequence⟩`

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 necessarly 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 footnotetest itself (used as target by `\footref`s 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@makefnma` 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@cfgpoint` (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@cfgpoint` and `\@makefntext@cfgpointii`. 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 `\@makefntext@cfgpoint` and `\@makefntext@cfgpointii`.
- memoir is not supported at all and errors when the code tries to patch `\@makefntext`.

To be documented

5 The Implementation

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

```
⟨*kernel⟩
⟨@@=fnote⟩
```

5.1 File declaration

```
\ProvidesFile{latex-lab-footnotes.ltx}
        [2023-02-04 v0.7 changes to the footnote interfaces]
```

5.2 code not fully handled yet

```
%
% latex.ltx
% not looked at yet
% \@mpfootnotetext is probably no longer needed, or only to support other
% classes and package. See below about the minipage code.
%
% \long\def\@mpfootnotetext#1{%
%  \global\setbox\@mpfootins\vbox{%
%    \unvbox\@mpfootins
%    \reset@font\footnotesize
%    \hsize\columnwidth
%    \@parboxrestore
%    \def\@currentcounter{mpfootnote}%
%    \protected@edef\@currentlabel
%         {\csname p@mpfootnote\endcsname\@thefnmark}%
%    \color@begingroup
%      \@makefntext{%
%        \rule\z@\footnotesep\ignorespaces#1\@finalstrut\strutbox}%
%    \par
%    \color@endgroup}}
% ========
% used by the minipage footnote code.
%
% \def\@mpfn{footnote}
% \def\thempfn{\thefootnote}
```

```
% =========
% this perhaps need some configuration options.
%
%\def\@makefnmark{\hbox{\@textsuperscript{\normalfont\@thefnmark}}}
%
% =========
%% alterations not covered:
%
% ./arabtex/afoot.sty  --- too different (and probably too old)
%
% =====
% alterations of footnotetext not covered:
%
% ./revtex4-1/revtex4-1.cls  ./revtex/ltxutil.sty ./revtex/revtex4-2.cls ... (need analysis)
% ./bigfoot/bigfoot.sty
%
% memoir needs checking too
%
% =====
%
% use of kerns to mark h-mode positions (unit sp)
%
% 1 = CJK
% 2 = CJK
% 3 = multiple footnotes (footmisc, koma, eledmac, tufte, memoir,
%    parnotes, sidenotes)
% 3 = outer kern in letter spacing (letterspace)
% 3 = beginning of list (examdesign.cls)
% 4 = CJK pigin
% 5 = CJK ruby

% 1-4 = polyglossia for korean
%
```

```
\ExplSyntaxOn
```

5.3 Temporary variables

```
\prop_new:N \l__fnote_tmpa_prop
\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`

```
\tl_new:N  \l_fnote_type_tl
\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

```
\bool_new:N       \l_fnote_link_bool
\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

```
\tl_new:N  \l_fnote_link_type_tl
\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

```
\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.

```
\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

```
\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

```
\bool_new:N       \l__fnote_autodetect_bool
\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.

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

5.6 Variants

```
\cs_generate_variant:Nn \ref_label:nn { Vn }
\cs_generate_variant:Nn \ref_value:nn { Vn }
\cs_generate_variant:Nn \prop_gput:Nnn {cxn}
\cs_generate_variant:Nn \hook_gput_code:nnn{nne}
\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.

```
\cs_new_protected:Npn \fnote_step_fnmark:nn #1#2 {
  \tl_if_novalue:nTF {#1}
    {
      \stepcounter {#2}
      \protected@xdef \@thefnmark { \use:c { the#2 } }
    }
    {
     \group_begin:
```

Note that this is a local assignment even though LaTeX 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.

```
        \int_set:cn { c@#2 }{ #1 }
        \unrestored@protected@xdef \@thefnmark { \use:c { the#2 } }
     \group_end:
    }
}
```

(*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.

```
\cs_new_protected:Npn \fnote_set_fnmark:nn #1#2 {
  \tl_if_novalue:nTF {#1}
    {
      \protected@xdef \@thefnmark { \use:c { the#2 } }
    }
    {
     \group_begin:
        \int_set:cn { c@#2 }{ #1 }
        \unrestored@protected@xdef \@thefnmark { \use:c { the#2 } }
     \group_end:
    }
}
```

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

5.8 Hooks

Hooks in the footnotemark command.

```
\NewMirroredHookPair{fnmark/before}{fnmark/after}
\NewHook{fnmark}
```

```
\NewHook{fnmark/begin}
\NewHook{fnmark/end}
```

Hooks in the footnotetext command

```
\NewMirroredHookPair{fntext/before}{fntext/after}
\NewHook{fntext}
\NewHook{fntext/para}
\NewHook{fntext/begin}
\NewHook{fntext/end}
```

5.9 Debugging code

the debugging code is just temporary

For now we have debugging turned on by default

```
\bool_new:N         \g_fnote_debug_bool
\bool_gset_true:N   \g_fnote_debug_bool
```

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

```
\cs_new_protected:Npn \__fnote_debug_footnotemark:
  {
    \bool_if:NT \g_fnote_debug_bool
       {
         \LogHook{fnmark/before}
         \LogHook{fnmark}
         \LogHook{fnmark/begin}
         \LogHook{fnmark/end}
         \LogHook{fnmark/after}
         \cs_gset_eq:NN \__fnote_debug_footnotemark: \prg_do_nothing:
       }
  }
```

Similar for the footnotetext

```
\cs_new_protected:Npn \__fnote_debug_footnotetext:
  {
    \bool_if:NT \g_fnote_debug_bool
       {
         \cs_log:N\@footnotetext@cfgpoint
         \cs_log:N\@footnotetext@cfgpointii
         \cs_log:N\@footnotetext@cfgpointiii
         \cs_log:N\@footnotetext@cfgpointiv
         \cs_log:N\@makefntext@cfgpoint
         \cs_log:N\@makefntext@cfgpointii
         \LogHook{fntext/before}
         \LogHook{fntext}
         \LogHook{fntext/para}
         \LogHook{fntext/begin}
         \LogHook{fntext/end}
         \LogHook{fntext/after}
```

Show the info only once (if at all).

```
         \cs_gset_eq:NN \__fnote_debug_footnotetext: \prg_do_nothing:
       }
  }
```

5.10 The new \@footnotemark command

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

```
\cs_new_protected:Npn \fnote_footnotemark: {
  \__fnote_debug_footnotemark:
%-------
% bibarts
% chextras  --- actually in the wrong place does an \unskip
  \UseHook{fnmark/before}
%-------
  \leavevmode
  \ifhmode
    \edef\@x@sf{\the\spacefactor}
%-------
% bxjsja-minimal.def   --- what they do could be done at ‘‘bibarts’’
%                          (a bit less efficient)
% memhfixc.sty
% footmisc.sty
    \UseHook{fnmark}
%-------
    \nobreak
  \fi
%-------
% hyperref.sty
  \UseHook{fnmark/begin}
%-------
  \@kernel@process@makefnmark
     \@makefnmark
%-------
```

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.

```
  \__fnote_footnotemark_finish:
}
```

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

__fnote_finish:
__fnote_footnotemark_finish:

```
\cs_new_protected:Npn \__fnote_finish: {
% hyperref.sty
% memhfixc.sty  --- could move fnmark/after
% scrlttr2.cls  --- could vanish if footmisc uses a hook
% footmisc.sty
  \UseHook{fnmark/end}
%-------
  \ifhmode
    \spacefactor \@x@sf \relax
  \fi
%
%-------
  \UseHook{fnmark/after}
```

```
%-------
}

\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

```
\def \@kernel@process@makefnmark { }
```

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

`\@footnotemark` At last provide the name LaTeX 2_ε is used to.

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

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

5.11 The new `\@footnotetext` command

`\fnote_footnotetext:n`

```
\cs_new_protected:Npn \fnote_footnotetext:n #1 {
  \__fnote_debug_footnotetext:
%-------
% ./linguex/linguex.sty
  \UseHook{fntext/before}
%-------
  \@kernel@process@footnotetext
  \@footnotetext@cfgpoint {  % config point
%-------
% resetting baselinestretch ... (could be done further down)
% ./uafthesis/uafthesis.cls
% ./setspace/setspace.sty
% ./footmisc/footmisc.sty (normal)
    \UseHook{fntext}
%-------
    \reset@font
    \footnotesize
%-------
% some classes use a different font size, e.g.,
% ./nrc/nrc1.cls  ./nrc/nrc2.cls
% but those could be done in fntext/para instead
%-------
```

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

```
    \interlinepenalty\interfootnotelinepenalty
    \splittopskip\footnotesep
    \splitmaxdepth \dp\strutbox
    \floatingpenalty \@MM
    \hsize\columnwidth
    \@parboxrestore
    \parindent 1em     % typical default used in \@makefntext moved up here
    \def\@currentcounter{footnote}
    \protected@edef \@currentlabel { \p@footnote \@thefnmark }
```

```
%-------
% for altering para parameters ...
% code for resphilosophica came earlier but it could go here.
% Has the advantage that one can also overwrite \cs{@currentcounter}
% and \cs{@currentlabel} is that is necessary.
%
% ./resphilosophica/resphilosophica.cls
    \UseHook{fntext/para}
%-------
    \color@begingroup
%-------
% fnpara wants to replace \@makefntext{...} and para and side option of footmisc etc too ...
% so we make this a config point
%-------
      \@footnotetext@cfgpointii       % config point
        {
%-------
% ./resphilosophica/resphilosophica.cls
%-------
          \@footnotetext@cfgpointiii  % config point
%-------
% bibarts
% fnbreak.sty
          \UseHook{fntext/begin}
%-------
          \ignorespaces
          #1
%-------
% bibarts
% fnbreak.sty
          \UseHook{fntext/end}
%-------
          \@footnotetext@cfgpointiv   % config point
        }
      \par
    \color@endgroup
  }
%-------
% ./linguex/linguex.sty
  \UseHook{fntext/after}
%-------
}
```

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

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

```
\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)

```
\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)

```
\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)

```
\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)

```
\cs_new_protected:Npn \@kernel@process@footnotetext {}
```

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

Provide the name LaTeX 2_ε is used to and do this unconditionally (no patching of class code if any). This means that if a class provides it own definition that gets lost and if necessary needs to be handled with firstaid (or updating of the class).

```
\AddToHook{begindocument}
  {
    \cs_set_eq:NN \@footnotetext \fnote_footnotetext:n
  }
```

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.

```
\newdimen\footnotemargin
\footnotemargin\maxdimen         % no value given

\AtBeginDocument
  {
    \ifdim \footnotemargin=\maxdimen
      \setlength\footnotemargin{1.8em}
    \fi
  }
```

\fnote_makefntext:n

```
\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.

```
  \@makefntext@cfgpoint
      {
        \ifdim\footnotemargin>\z@
          \hb@xt@ \footnotemargin{\hss\@makefnmark}
        \else
          \ifdim\footnotemargin=\z@
            \llap{\@makefnmark}
          \else
          \ifdim\footnotemargin=-\maxdimen
               \@makefnmark
            \else
               \llap{\hb@xt@ -\footnotemargin{\@makefnmark\hss}}
```

```
            \fi
          \fi
        \fi
      }
  \@makefntext@cfgpointii
      { #1 }
}
```

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

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

```
\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)

```
\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`

```
\newcommand\old@std@class@makefntext[1]{%
    \parindent 1em%
    \noindent
    \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).

```
\tl_new:N \l__fnote_patch_tl
\cs_new_eq:NN \__fnote_tmp:w \ERROR

\cs_new_protected:Npn \__fnote_patch:
  {
    \tl_set:No \l__fnote_patch_tl { \@makefntext { \@makefntext@cfgpointii{##1} } }
    \tl_if_in:NnTF \l__fnote_patch_tl { \hbox }
      { \cs_set_eq:NN \__fnote_tmp:w \__fnote_patch_hbox:w }
      {
        \tl_if_in:NnTF \l__fnote_patch_tl { \hb@xt@ }
          { \cs_set_eq:NN \__fnote_tmp:w \__fnote_patch_hb@xt@:w }
          {
            \tl_if_in:NnTF \l__fnote_patch_tl { \@makefnmark }
              { \cs_set_eq:NN \__fnote_tmp:w \__fnote_patch_@makefnmark:w }
              { \ERROR
                \cs_set_eq:NN \__fnote_tmp:w \exp_stop_f: }
          }
      }
    \tl_set:Nf \l__fnote_patch_tl
      { \exp_after:wN \__fnote_tmp:w \l__fnote_patch_tl }
    \cs_set:Npn \__fnote_tmp:w { \long \def \@makefntext ####1 }
    \exp_after:wN \__fnote_tmp:w \exp_after:wN { \l__fnote_patch_tl }
  }
```

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

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

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

Same for the other cases.

```
\cs_new:Npn \__fnote_patch_hb@xt@:w #1 \hb@xt@ #2 #
  { \exp_stop_f: #1 \@makefntext@processX { \hb@xt@ #2 } }
\cs_new:Npn \__fnote_patch_@makefnmark:w #1 \@makefnmark
  { \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:

```
\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.

```

\AddToHook{begindocument}
  {
    \cs_if_eq:NNTF \@makefntext \old@std@class@makefntext
      {
        \cs_set_eq:NN \@makefntext \fnote_makefntext:n
      }
```

```
      {
        \__fnote_patch:
      }
  }


% possibly add the following to check for multiple \hbox in
% the definition:
%
% \seq_set_split:NnV \l__fnote_patch_seq { \hbox } \l__fnote_patch_tl
% \int_compare:nT { \seq_count:N \l__fnote_patch_seq } > 2 \ERROR
%
```

5.13 Document-level commands

\footnotetext

```
\DeclareDocumentCommand\footnotetext {om}
  {
    \fnote_set_fnmark:nn {#1} \@mpfn
    \@footnotetext {#2}
  }
```

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

\footnote

```
\DeclareDocumentCommand\footnote {om}
  {
    \fnote_step_fnmark:nn {#1} \@mpfn
    \cs_set_eq:NN \__fnote_footnotemark_finish: \prg_do_nothing:
    \@footnotemark
    \cs_set_eq:NN \__fnote_footnotemark_finish: \__fnote_finish:
    \@footnotetext {#2}
    \__fnote_footnotemark_finish:
  }
```

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

\footnotemark

```
\DeclareDocumentCommand\footnotemark {o}
  {
    \fnote_step_fnmark:nn {#1} { footnote }
    \@footnotemark
  }
```

(*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`.

```
\DeclareDocumentCommand\footref {m}
  {
    \begingroup
      \unrestored@protected@xdef\@thefnmark{\ref*{#1}}%
```

```
    \endgroup
    \bool_set_false:N  \l__fnote_autodetect_bool
    \tl_set:Nn \l__fnote_currentlabel_tl {#1}
    \@footnotemark
    \bool_set_true:N  \l__fnote_autodetect_bool
  }
```

(*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.

```
\cs_new_protected:Npn \@mpfootnotetext@cfgpoint #1
 {
   \global\setbox\@mpfootins\vbox{%
    \unvbox\@mpfootins
    #1
    }
 }
```

Then we patch `\@iiiminipage`

```
\RequirePackage{etoolbox}
\ExplSyntaxOff
\patchcmd\@iiiminipage
 {\let\@footnotetext\@mpfootnotetext}
 {\let\@footnotetext@cfgpoint\@mpfootnotetext@cfgpoint}
 {}{\ERROR}
\ExplSyntaxOn
```

5.15.1 memoir

The memoir class redefines various internal commands to inject its hooks and addtional 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.

```
\AddToHook{class/memoir/before}
  { \let\new@std@class@makecol\@makecol }
\AddToHook{class/memoir/after}
  {
    \cs_set_eq:NN  \@footnotemark \fnote_footnotemark:
    \cs_set_eq:NN  \@makefntext\old@std@class@makefntext
    \cs_set_eq:NN  \@makecol\new@std@class@makecol
  }
```

5.15.2 setspace

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

```
\AddToHook{package/setspace/after}
   {\let \@footnotetext \fnote_footnotetext:n
    \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.

```
\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.

```
\AtBeginDocument
  {
   \providecommand\hyper@linkstart{\@gobbletwo}
   \providecommand\hyper@linkend{\@empty}
  }
```

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).

```
\AtBeginDocument
 {
   \@ifpackageloaded{hyperref}
     {
       \legacy_if:nF{Hy@hyperfootnotes}{\bool_set_false:N \l_fnote_link_bool}
     }
     {
       \bool_set_false:N \l_fnote_link_bool
     }
 }
```

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`

```
\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:`

```
\cs_new_protected:Npn \__fnote_label_hook: {}
\AddToHook{begindocument/before}
  {
    \RequirePackage{nameref}
    \g@addto@macro\label@hook
      {
        \__fnote_label_hook:
```

```
      }
  }
```

(*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`.

```
\AddToHook{fntext/begin}
 {
   \cs_set_protected:Npn \__fnote_label_hook:
    {
      \ref_label:Vn \label@name {fnote/struct}
    }
 }
```

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!

```
\cs_new_protected:Npn \fnote_new:nn #1 #2 % #1 name, #2 options
 {
   \prop_new:c { g__fnote_currentmarks_ #1 _prop }
 }

\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)

```
\cs_new_protected:Npn \fnote_mark_gput:nn #1 #2 % #1 the representation of the mark, #2 type
 {
   \prop_gput:cxn { g__fnote_currentmarks_ #2 _prop }
     { \tag_get:n{struct_num} }
     { #1 }
 }

\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

```
\cs_new_protected:Npn \fnote_mark_gpop:nnN #1 #2 #3
  {
    \seq_clear:N #3
    \prop_set_eq:Nc     \l__fnote_tmpa_prop { g__fnote_currentmarks_ #2 _prop }
```

```
    \prop_map_inline:Nn \l__fnote_tmpa_prop
      {
        \tl_if_eq:nnT {#1} { ##2 }
          {
```

store the key (the structure number) in the seq

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

remove entry as used from the global prop

```
            \prop_gremove:cn { g__fnote_currentmarks_ #2 _prop } {##1}
          }
      }
  }
\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 `\@makefntext`. At this time `\@thefnmark` is already set.

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

end an open mc and start the structure.

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

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

```
     \bool_if:NTF \l__fnote_autodetect_bool
      {
```

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.

```
        \fnote_mark_gput:oo {\@thefnmark}{\l_fnote_type_tl}
        \tl_set:Nx \l__fnote_linktarget_tl {footnote*.\tag_get:n{struct_num}}
      }
```

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.

```
      {
        \hook_gput_code:nne {tagpdf/finish/before} {tagpdf/footnote}
         {
           \exp_not:N\fnote_gput_refs:ee
            { \tag_get:n{struct_num} }
            { \ref_value:Vn\l__fnote_currentlabel_tl {fnote/struct} }
         }
```

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

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

And now the actual content

```
      \tag_mc_begin:n{tag=Lbl}
      %
      \bool_if:NTF \l_fnote_link_bool
       {
         \exp_args:No
            \hyper@linkstart
            { \l_fnote_link_type_tl }
            { \l__fnote_linktarget_tl }
            #1
            \hyper@linkend
       }
       { #1 }
     \tag_mc_end:
     \tag_struct_end:
     \tag_mc_begin_pop:n{}
   }
```

At last set the kernel command to this one

```
\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

```
\cs_new_protected:Npn \__fnote_gput_ref:nn #1 #2 %#1 the structure number receiving the ref
  {
    \tag_if_active:T
      {
        \tag_struct_gput:nne {#1}{ref}{\tag_struct_object_ref:e { #2 }}
      }
  }
\cs_new_protected:Npn \fnote_gput_refs:nn #1 #2 % pair of numbers
  {
    \__fnote_gput_ref:nn {#1}{#2}
    \__fnote_gput_ref:nn {#2}{#1}
  }
\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* ??.)

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

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

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

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

```
    \tag_check_child:nnTF {FENote}{pdf2}
      {
        \tag_struct_begin:n { tag=footnote }
      }
```

```
      {
        \tag_struct_begin:n { tag=footnote,parent=\l__fnote_dflt_struct_tl }
      }
```

Store the current structure number for labels.

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

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

```
    \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.

```
    \bool_if:NTF \l__fnote_autodetect_bool
      {
```

find open marks with identical `\@thefnmark`

```
        \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:

```
       \seq_map_inline:Nn \l__fnote_currentrefs_seq
         {
           \fnote_gput_refs:ee {##1}{ \l__fnote_currentstruct_tl }
         }
      }
```

If no auto detection is done

```
      {%no auto

      }
```

This finish the setup of the tagging structure. Now we process the text. The destinations for the links are set with the label so that we can be sure that we are in hmode.

```
      #1 {#2}
    \tag_struct_end:
    \tag_mc_begin_pop:n{}
  }
```

At last set the kernel command to this one

```
\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??

```
\cs_set_protected:Npn \@makefntext@cfgpoint #1 %#1 code that typesets the mark.
  {
    \noindent
    \tag_mc_end_push:
    \tag@FENoteLbl { #1 }
    \tag_mc_begin_pop:n{}
  }
```

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

\tag@FENoteLbl

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

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)

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

```
    \tag_struct_begin:n { tag=NonStruct,parent=\l__fnote_currentstruct_tl +1 }
     \tag_mc_begin:n { tag=Lbl }
      #1
     \tag_mc_end:
    \tag_struct_end:
  }
```

(*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

```
\cs_set_protected:Npn \@makefntext@cfgpointii #1
  {
    \tag_mc_end_push:
    \tag@FENotetext { #1 }
    \tag_mc_begin_pop:n{}
  }
```

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

```
\cs_new_protected:Npn \tag@FENotetext #1
  {
    \tag_mc_begin:n{}
    #1
    \tag_mc_end:
  }
```

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

```
\ExplSyntaxOff
```

⟨/kernel⟩

⟨@@=⟩

6 Reimplementing the **footmisc** package

```
⟨*footmisc⟩
%%
%% Copyright (c) 1995-2011 Robin Fairbairns
%% Copyright (c) 2018-2022 Robin Fairbairns, Frank Mittelbach
%%
%% This file is part of the 'latex-lab Bundle'.
%% -------------------------------------------
%%
%% It may be distributed and/or modified under the
%% conditions of the LaTeX Project Public License, either version 1.3c
%% of this license or (at your option) any later version.
%% The latest version of this license is in
%%    https://www.latex-project.org/lppl.txt
%% and version 1.3c or later is part of all distributions of LaTeX
%% version 2008 or later.
%%
%% This work has the LPPL maintenance status 'maintained'.
%%
%%
%% File: footmisc.dtx (C) Copyright 1995-2011 Robin Fairbairns
%%                    (C) Copyright 2018-2022 Frank Mittelbach
\NeedsTeXFormat{LaTeX2e}
\providecommand\DeclareRelease[3]{}
\providecommand\DeclareCurrentRelease[2]{}

\DeclareRelease{v5}{2011-06-06}{footmisc-2011-06-06.sty}
\DeclareCurrentRelease{}{2022-02-14}
\ProvidesPackage{latex-lab-footmisc}%
        [2022/03/08 v6.0d
     a miscellany of footnote facilities -- latex-lab version%
                  ]

\NeedsTeXFormat{LaTeX2e}[2020/10/01]
\newtoks\FN@temptoken
\providecommand\protected@writeaux{%
  \protected@write\@auxout
}
\def\l@advance@macro{\@@dvance@macro\edef}
\def\@@dvance@macro#1#2#3{\expandafter\@tempcnta#2\relax
  \advance\@tempcnta#3\relax
  #1#2{\the\@tempcnta}%
}
\let\@advance@macro\l@advance@macro
\DeclareOption{symbol}{\renewcommand\thefootnote{\fnsymbol{footnote}}}
\newif\ifFN@robust \FN@robustfalse
\DeclareOption{symbol*}{%
  \renewcommand\thefootnote{\@fnsymbol\c@footnote}%
  \FN@robusttrue
  \AtEndOfPackage{\setfnsymbol{lamport*-robust}}%
}
\newif\ifFN@para  \FN@parafalse
\DeclareOption{para}{%
```

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

```
%    \PackageError{footmisc}{Option "\CurrentOption" incompatible with
%                            option "side"}%
%                 {I shall ignore "\CurrentOption"}%
  \FN@paratrue
  \setlength\footnotemargin{-\maxdimen}    % default when para is used
}
\DeclareOption{side}{\ifFN@para
    \PackageError{footmisc}{Option "\CurrentOption" incompatible with
                            option "para"}%
                {I shall ignore "\CurrentOption"}%
  \else
    \def\@footnotetext@cfgpoint {\marginpar}
    \AddToHook{fntext/para}{%
      \hsize\marginparwidth     % correct the default \hsize
      \footnotesep\z@           % don't add a default separation
    }
    \def\@footnotetext@cfgpointii  {\@makefntext}
    \def\@footnotetext@cfgpointiii {}
    \def\@footnotetext@cfgpointiv  {}
  \fi
}
\let\footnotelayout\@empty
\DeclareOption{ragged}{%
  \@ifundefined{RaggedRight}%
    {\renewcommand\footnotelayout{\linepenalty50 \raggedright}}%
    {\renewcommand\footnotelayout{\linepenalty50 \RaggedRight}}%
}
\newif\ifFN@perpage
\FN@perpagefalse
\DeclareOption{perpage}{%
  \FN@perpagetrue
}
\newif\ifFN@fixskip      \FN@fixskipfalse

\let\FN@bottomcases\thr@@
\newif\ifFN@abovefloats  \FN@abovefloatstrue
\DeclareOption{bottom}{%
  \let\FN@bottomcases\@ne
  \FN@abovefloatsfalse
  \FN@fixskiptrue
}
\DeclareOption{bottomfloats}{%
  \let\FN@bottomcases\tw@
  \FN@abovefloatstrue \FN@fixskiptrue
}
\DeclareOption{abovefloats}{\FN@abovefloatstrue  \FN@fixskiptrue}
\DeclareOption{belowfloats}{\FN@abovefloatsfalse \FN@fixskiptrue}
\DeclareOption{marginal}{%
  \footnotemargin-0.8em\relax
}
\DeclareOption{flushmargin}{%
  \footnotemarginOpt\relax
```

```
}
\newif\ifFN@hangfoot  \FN@hangfootfalse
\DeclareOption{hang}{%
  \FN@hangfoottrue
}
\newcommand*\hangfootparskip{0.5\baselineskip}
\newcommand*\hangfootparindent{0em}%
\DeclareOption{norule}{%
  \renewcommand\footnoterule{}%
  \advance\skip\footins 4\p@\@plus2\p@\relax
}
\DeclareOption{splitrule}{%
  \gdef\split@prev{0}
  \let\pagefootnoterule\footnoterule
  \let\mpfootnoterule\footnoterule
  \def\splitfootnoterule{\kern-3\p@ \hrule \kern2.6\p@}
  \def\footnoterule{\relax
    \ifx \@listdepth\@mplistdepth
      \mpfootnoterule
    \else
      \ifnum\split@prev=\z@
        \pagefootnoterule
      \else
        \splitfootnoterule
      \fi
      \xdef\split@prev{\the\insertpenalties}%
    \fi
  }%
}
\newif\ifFN@stablefootnote  \FN@stablefootnotefalse
\DeclareOption{stable}{\FN@stablefootnotetrue}
\newif\ifFN@multiplefootnote  \FN@multiplefootnotefalse
\DeclareOption{multiple}{\FN@multiplefootnotetrue}
\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.

```
%\@ifundefined{@kernel@before@cclv}
%  {\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).

```
\ifFN@para
  \def\@makecol@cfgpointii {%
     \global\setbox\footins\vbox{\FN@makefootnoteparagraph}%
    }
\fi
\ifFN@fixskip
  \def\@outputbox@removebskip{%
    \ifx\@textbottom\relax \else
      \@outputbox@append{%
        \@tempskipa\lastskip
        \ifnum \gluestretchorder\@tempskipa>\z@
```

```
          \vskip-\@tempskipa
          \xdef\@outputbox@reinsertbskip
              {\noexpand\@outputbox@append{\vskip\the\@tempskipa}}%
        \else
          \global\let\@outputbox@reinsertbskip\relax
        \fi
      }%
   \fi
  }
\let\@outputbox@reinsertbskip\relax
\else
  \let\@outputbox@removebskip \relax
  \let\@outputbox@reinsertbskip\relax
\fi
\ifcase \FN@bottomcases\relax
\ERROR
\or
  \ifFN@abovefloats
    \def\@makecol@cfgpoint {%
       \@if@footnotes@TF
          {\@outputbox@append{\vfill}}%
          {\@if@bfloats@TF{\@outputbox@append{\vfill}}%
                          {\@outputbox@reinsertbskip}}%
       \@outputbox@appendfootnotes
       \@outputbox@attachfloats
     }
  \else
    \def\@makecol@cfgpoint {%
       \@outputbox@attachfloats
       \@if@footnotes@TF
          {\@outputbox@append{\vfill}}%
          {\@outputbox@reinsertbskip}%
       \@outputbox@appendfootnotes
    }
  \fi
\or
  \ifFN@abovefloats
     \def\@makecol@cfgpoint {%
        \@outputbox@appendfootnotes
        \@if@bfloats@TF
            {\@outputbox@append{\vfill}}%
            {\@outputbox@reinsertbskip}%
        \@outputbox@attachfloats
     }
  \else
     \def\@makecol@cfgpoint {%
       \@if@footnotes@TF
          {\@outputbox@append{\vfill}}%
          {\@if@bfloats@TF{\@outputbox@append{\vfill}}%
                          {\@outputbox@reinsertbskip}}%
       \@outputbox@attachfloats
       \@outputbox@appendfootnotes
     }
  \fi
```

```
\or
  \ifFN@abovefloats
    \def\@makecol@cfgpoint {%
       \@outputbox@appendfootnotes
       \@outputbox@attachfloats
       \@outputbox@reinsertbskip
    }
  \else
    \def\@makecol@cfgpoint {%
       \@outputbox@attachfloats
       \@outputbox@appendfootnotes
       \@outputbox@reinsertbskip
}
  \fi
\else
\ERROR
\fi

% next can be dropped when cleaned up
\newif\ifFN@setspace
\@ifpackageloaded{setspace}%
 {%
   \FN@setspacetrue
   \@ifclassloaded{memoir}%
     {%
       \AddToHook{fntext}{\let\baselinestretch\m@m@singlespace}%
       \let\FN@baselinestretch\m@m@singlespace
     }%
     {%
%       \AddToHook{fntext}{\let\baselinestretch\setspace@singlespace}%
       \let\FN@baselinestretch\setspace@singlespace
     }%
 }%
 {%
   \FN@setspacefalse
 }



\ifFN@para
  \def\@footnotetext@cfgpoint {\insert\footins}

  \long\def\@footnotetext@cfgpointii #1{%
    \setbox\FN@tempboxa\hbox{\@makefntext{#1}}%
    \dp\FN@tempboxa\z@
    \ht\FN@tempboxa
      \dimexpr\wd\FN@tempboxa *%
               \footnotebaselineskip /\columnwidth\relax
    \box\FN@tempboxa
  }


  \def\@footnotetext@cfgpointiii {}
  \def\@footnotetext@cfgpointiv {% config point
```

```
            \strut
            \penalty-10\relax
            \hskip\footglue
  }
\fi



\ifFN@para
  \let\FN@tempboxa\@tempboxa
  \newbox\FN@tempboxb
  \newbox\FN@tempboxc
  \newskip\footglue \footglue=1em plus.3em minus.3em

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
  \newdimen\footnotebaselineskip

  % establish late:

\AddToHook{begindocument/before} {%
  {%
    \footnotesize
    \global\footnotebaselineskip=\normalbaselineskip
  }%
}
```

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

```

  \long\def\FN@makefootnoteparagraph{%
    \FN@setfootnoteparawidth
    \@parboxrestore
    \baselineskip=\footnotebaselineskip
    \unvbox\footins \FN@removehboxes
    \RawParEnd
  }
  \def\FN@removehboxes{\setbox\FN@tempboxa\lastbox
    \ifhbox\FN@tempboxa{\FN@removehboxes}%
      \unhbox\FN@tempboxa
    \else
      \RawNoindent
      \rule\z@\footnotesep
    \fi
  }
\fi


\@ifpackageloaded{multicol}
  {\def\FN@setfootnoteparawidth
    {\hsize\ifnum\doublecol@number>\@ne
                  \textwidth
            \else \columnwidth \fi}}
  {\def\FN@setfootnoteparawidth{\hsize\columnwidth}}

\ifFN@perpage
```

```
\RequirePackage{perpage}
\MakePerPage{footnote}
```

Fix a bug in perpage ...

```
\def\@stpelt#1{\global\csname c@#1\endcsname \m@ne
  \stepcounter{#1}%
  \pp@fix@MakePerPage{#1}%
}
\def\pp@fix@MakePerPage#1{%
    \ifnum \value{#1}>\z@
      \addtocounter{#1}\m@ne\fi
}
```

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:

```
  \def\pp@cl@end@iii\stepcounter#1\pp@fix@MakePerPage#2{}
\fi


\ifFN@para

% This can use the default interface, except that a negative value for
% \footnotemargin makes little sense, so we test for this and warn if
% necessary. But -\maxdimen is ok again, so would need to be a litte bit more elaborate.
%

%\AddToHook{fntext/para}{
%  \ifdim \footnotemargin >\z@ \else
%    \PackageWarningNoline{footmisc}{Option 'para' needs positive \noexpand\footnotemargin}%
%    \footnotemargin 1.8em\relax
%  \fi
%}


\AddToHook{fntext/begin}{\nobreak \hspace{.2em}}




\else

  \ifFN@hangfoot
    \long\def\@makefntext#1{%
      \bgroup
        \setbox\@tempboxa\hbox{%
          \ifdim\footnotemargin>\z@
            \hb@xt@\footnotemargin{\@makefnmark\hss}%
          \else
            \@makefnmark
          \fi
```

```
        }%
        \leftmargin\wd\@tempboxa
        \rightmargin\z@
        \linewidth \columnwidth
        \advance \linewidth -\leftmargin
        \parshape \@ne \leftmargin \linewidth
        \footnotesize
        \@setpar{{\@@par}}%
        \leavevmode
        \llap{\box\@tempboxa}%
        \parskip\hangfootparskip\relax
        \parindent\hangfootparindent\relax
        \footnotelayout#1%
        \par
      \egroup
    }

 \else

% This is now using the default interface:
%
% \long\def\@makefntext#1{%
%      \parindent1em
%      \noindent
%      \ifdim\footnotemargin>\z@
%        \hb@xt@ \footnotemargin{\hss\@makefnmark}%
%      \else
%        \ifdim\footnotemargin=\z@
%          \llap{\@makefnmark}%
%        \else
%          \llap{\hb@xt@ -\footnotemargin{\@makefnmark\hss}}%
%        \fi
%      \fi
%    \footnotelayout#1%
%  }

 \fi
\fi




\ifFN@multiplefootnote
  \providecommand*{\multiplefootnotemarker}{3sp}
% we tag the separator as artifact
  \ExplSyntaxOn
  \providecommand*{\multfootsep}{\tag_mc_end_push:\tag_mc_begin:n{artifact},\tag_mc_end:\tag
  \ExplSyntaxOff
  \AddToHook{fnmark}      {\FN@mf@check}
  \AddToHook{fnmark/end}  {\FN@mf@prepare}
%
  \def\FN@mf@prepare{%
    \kern-\multiplefootnotemarker
    \kern\multiplefootnotemarker\relax
```

```
  }
  \def\FN@mf@check{%
    \ifdim\lastkern=\multiplefootnotemarker\relax
%?? is that necessary or even correct ??
      \edef\@x@sf{\the\spacefactor}%
%?? shouldn't that be 2 unkerns ?? (none would also be ok)
      \unkern  % new
      \unkern
      \textsuperscript{\multfootsep}%
      \spacefactor\@x@sf\relax
    \fi
  }
\else
  \let\FN@mf@prepare\relax
\fi
\ifFN@stablefootnote
\let\FN@sf@@footnote\footnote
\def\footnote{\ifx\protect\@typeset@protect
    \expandafter\FN@sf@@footnote
  \else
    \expandafter\FN@sf@gobble@opt
  \fi
}
\edef\FN@sf@gobble@opt{\noexpand\protect
  \expandafter\noexpand\csname FN@sf@gobble@opt \endcsname}
\expandafter\def\csname FN@sf@gobble@opt \endcsname{%
  \@ifnextchar[%]
    \FN@sf@gobble@twobracket
    \@gobble
}
\def\FN@sf@gobble@twobracket[#1]#2{}
\let\FN@sf@@footnotemark\footnotemark
\def\footnotemark{\ifx\protect\@typeset@protect
    \expandafter\FN@sf@@footnotemark
  \else
    \expandafter\FN@sf@gobble@optonly
  \fi
}
\edef\FN@sf@gobble@optonly{\noexpand\protect
  \expandafter\noexpand\csname FN@sf@gobble@optonly \endcsname}
\expandafter\def\csname FN@sf@gobble@optonly \endcsname{%
  \@ifnextchar[%]
    \FN@sf@gobble@bracket
    {}%
}
\def\FN@sf@gobble@bracket[#1]{}
\fi
\newcommand\setfnsymbol[1]{%
  \@bsphack
  \@ifundefined{FN@fnsymbol@#1}%
  {%
    \PackageError{footmisc}{Symbol style "#1" not known}%
    \@eha
  }{%
```

```
    \expandafter\let\expandafter\@fnsymbol\csname
                           FN@fnsymbol@#1\endcsname
  }%
  \@esphack
}
\let\FN@fnsymbol@lamport\@fnsymbol
\newif\if@tempswb
\DeclareDocumentCommand\DefineFNsymbols {smO{text}m}{%
  \expandafter\ifx\csname FN@fnsymbol@#2\endcsname\relax
    \PackageInfo{footmisc}{Declaring symbol style #2}%
  \else
    \PackageWarning{footmisc}{Redeclaring symbol style #2}%
  \fi
  \toks@{}%
  \def\@tempb{\end}%
  \FN@build@symboldef#4\end
  \def\@tempc{math}%
  \def\@tempd{#3}%
  \expandafter\xdef\csname FN@fnsymbol@#2\endcsname##1{%
    \ifx\@tempc\@tempd
      \noexpand\ensuremath
    \else
      \noexpand\nfss@text
    \fi
    {%
      \noexpand\ifcase##1%
      \the\toks@
      \noexpand\else
      \IfBooleanTF#1{\noexpand\@ctrerr}%
        {\noexpand\FN@orange##1}%
      \noexpand\fi
    }%
  }%
}
\def\FN@build@symboldef#1{%
  \def\@tempa{#1}%
  \ifx\@tempa\@tempb
  \else
    \toks@\expandafter{\the\toks@\or#1}%
    \expandafter\FN@build@symboldef
  \fi
}
\DeclareDocumentCommand\DefineFNsymbolsTM {smm}{%
  \expandafter\ifx\csname FN@fnsymbol@#2\endcsname\relax
    \PackageInfo{footmisc}{Declaring symbol style #2}%
  \else
    \PackageWarning{footmisc}{Redeclaring symbol style #2}%
  \fi
  \toks@{}%
  \def\@tempb{\end}%
  \FN@build@symboldefTM#3\end\@null
  \expandafter\xdef\csname FN@fnsymbol@#2\endcsname##1{%
    \noexpand\ifcase##1%
      \the\toks@
```

```
    \noexpand\else
      \IfBooleanTF#1{\noexpand\@ctrerr}%
        {\noexpand\FN@orange##1}%
      \noexpand\fi
  }%
}
\def\FN@build@symboldefTM#1#2{%
  \def\@tempa{#1}%
  \ifx\@tempa\@tempb
  \else
    \toks@\expandafter{\the\toks@\or\TextOrMath{#1}{#2}}%
    \expandafter\FN@build@symboldefTM
  \fi
}
\def\FN@orange#1{%
  \ifFN@robust
    \@arabic#1%
    \@bsphack
    \PackageInfo{footmisc}{Footnote number \number#1 out of range}%
    \protect\@fnsymbol@orange
    \@esphack
  \else \@ctrerr \fi
}
\global\let\@diagnose@fnsymbol@orange\relax
\AtEndDocument{\@diagnose@fnsymbol@orange}
\def\@fnsymbol@orange{%
  \gdef\@diagnose@fnsymbol@orange{%
    \PackageWarningNoLine{footmisc}{Some footnote number(s)
      were out of range
      \MessageBreak
      see log for details%
    }%
  }%
}
\DefineFNsymbolsTM{bringhurst}{%
  \textasteriskcentered *%
  \textdagger    \dagger
  \textdaggerdbl \ddagger
  \textsection   \mathsection
  \textbardbl    \|%
  \textparagraph \mathparagraph
}%
\DefineFNsymbolsTM{chicago}{%
  \textasteriskcentered *%
  \textdagger    \dagger
  \textdaggerdbl \ddagger
  \textsection   \mathsection
  \textbardbl    \|%
  \#\#%
}%
\DefineFNsymbolsTM{wiley}{%
  \textasteriskcentered *%
  {\textasteriskcentered\textasteriskcentered}{**}%
  \textdagger    \dagger
```

```
  \textdaggerdbl \ddagger
  \textsection   \mathsection
  \textparagraph \mathparagraph
  \textbardbl    \|%
}%
\DefineFNsymbolsTM{lamport-robust}{%
  \textasteriskcentered *%
  \textdagger    \dagger
  \textdaggerdbl \ddagger
  \textsection   \mathsection
  \textparagraph \mathparagraph
  \textbardbl    \|%
  {\textasteriskcentered\textasteriskcentered}{**}%
  {\textdagger\textdagger}{\dagger\dagger}%
  {\textdaggerdbl\textdaggerdbl}{\ddagger\ddagger}%
}
\DefineFNsymbolsTM*{lamport*}{%
  \textasteriskcentered *%
  \textdagger    \dagger
  \textdaggerdbl \ddagger
  \textsection   \mathsection
  \textparagraph \mathparagraph
  \textbardbl    \|%
  {\textasteriskcentered\textasteriskcentered}{**}%
  {\textdagger\textdagger}{\dagger\dagger}%
  {\textdaggerdbl\textdaggerdbl}{\ddagger\ddagger}%
  {\textsection\textsection}{\mathsection\mathsection}%
  {\textparagraph\textparagraph}{\mathparagraph\mathparagraph}%
  {\textasteriskcentered\textasteriskcentered\textasteriskcentered}{***}%
  {\textdagger\textdagger\textdagger}{\dagger\dagger\dagger}%
  {\textdaggerdbl\textdaggerdbl\textdaggerdbl}{\ddagger\ddagger\ddagger}%
  {\textsection\textsection\textsection}%%
    {\mathsection\mathsection\mathsection}%
  {\textparagraph\textparagraph\textparagraph}%%
    {\mathparagraph\mathparagraph\mathparagraph}%
}
\setfnsymbol{lamport*}
\DefineFNsymbolsTM{lamport*-robust}{%
  \textasteriskcentered *%
  \textdagger    \dagger
  \textdaggerdbl \ddagger
  \textsection   \mathsection
  \textparagraph \mathparagraph
  \textbardbl    \|%
  {\textasteriskcentered\textasteriskcentered}{**}%
  {\textdagger\textdagger}{\dagger\dagger}%
  {\textdaggerdbl\textdaggerdbl}{\ddagger\ddagger}%
  {\textsection\textsection}{\mathsection\mathsection}%
  {\textparagraph\textparagraph}{\mathparagraph\mathparagraph}%
  {\textasteriskcentered\textasteriskcentered\textasteriskcentered}{***}%
  {\textdagger\textdagger\textdagger}{\dagger\dagger\dagger}%
  {\textdaggerdbl\textdaggerdbl\textdaggerdbl}{\ddagger\ddagger\ddagger}%
  {\textsection\textsection\textsection}%%
    {\mathsection\mathsection\mathsection}%
```

```
  {\textparagraph\textparagraph\textparagraph}%%
    {\mathparagraph\mathparagraph\mathparagraph}%
}
\newcommand\mpfootnotemark{%
  \@ifnextchar[%
    \@xmpfootnotemark
    {%
      \stepcounter\@mpfn
      \protected@xdef\@thefnmark{\thempfn}%
      \@footnotemark
    }%
}
\def\@xmpfootnotemark[#1]{%
  \begingroup
    \csname c@\@mpfn\endcsname #1\relax
    \unrestored@protected@xdef\@thefnmark{\thempfn}%
  \endgroup
  \@footnotemark
}
```

TEMP PATCHES FOR TESTING

```
\endinput
```

⟨/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
\| 1147, 1155, 1165, 1173, 1184, 1205

A

\addtocounter 909
\AddToHook 276, 346, 407, 409, 415, 417, 436, 444, 654, 820, 824, 868, 922, 930, 994, 995
\advance 629, 703, 950
\AtBeginDocument 283, 419, 424
\AtEndDocument 1132
\AtEndOfPackage 638

B

\baselineskip 699, 878
\baselinestretch 417, 820, 824
\begingroup 385, 1229
\bgroup 939
bool commands:
  \bool_gset_true:N 119
  \bool_if:NTF . . 122, 134, 483, 499, 541
  \bool_new:N 68, 75, 118
  \bool_set_false:N 388, 428, 431
  \bool_set_true:N 69, 76, 391
\box 843, 955

C

\columnwidth . . 15, 224, 842, 897, 898, 949
\cs 232, 233
cs commands:
  \cs_generate_variant:Nn 79, 80, 81, 82, 83, 464, 478, 526
  \cs_gset_eq:NN 129, 148
  \cs_if_eq:NNTF 348
  \cs_log:N . . 136, 137, 138, 139, 140, 141
  \cs_new:Npn 338, 340, 342, 344
  \cs_new_eq:NN 195, 316
  \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
  \cs_set:Npn 335

\cs_set_eq:NN 197, 278, 322, 325, 328, 330, 350, 372, 374, 411, 412, 413, 513, 556
\cs_set_protected:Npn . 446, 557, 575
\csname 19, 903, 1024, 1025, 1039, 1040, 1054, 1062, 1072, 1097, 1105, 1230
\CurrentOption 642, 644, 649, 651

D

\dagger . . 1144, 1152, 1161, 1169, 1175, 1180, 1186, 1191, 1201, 1207, 1212
\ddagger . 1145, 1153, 1162, 1170, 1176, 1181, 1187, 1192, 1202, 1208, 1213
\DeclareCurrentRelease 613, 616
\DeclareDocumentCommand 364, 369, 378, 383, 1061, 1096
\DeclareOption 633, 635, 641, 648, 664, 671, 678, 683, 687, 688, 689, 692, 696, 701, 705, 724, 726
\DeclareRelease 612, 615
\def 11, 17, 28, 29, 33, 196, 227, 335, 418, 627, 628, 653, 658, 659, 660, 709, 710, 731, 736, 759, 768, 778, 786, 797, 803, 835, 837, 847, 848, 875, 882, 894, 898, 903, 907, 911, 938, 967, 997, 1001, 1017, 1025, 1030, 1032, 1040, 1045, 1068, 1070, 1071, 1088, 1089, 1103, 1114, 1115, 1122, 1133, 1228
\DefineFNsymbols 1061
\DefineFNsymbolsTM 1096, 1142, 1150, 1158, 1167, 1178, 1199
\dimexpr 841
\dp 222, 839

E

\edef 160, 627, 1004, 1023, 1038
\egroup 960
\else 294, 297, 300, 652, 713, 716, 737, 744, 751, 767, 785, 802, 809, 885, 897, 923, 935, 943, 963, 972, 975, 1012, 1019, 1034, 1064, 1075, 1081, 1091, 1099, 1108, 1117, 1129
\end 1068, 1069, 1103, 1104
\endcsname 19, 903, 1024, 1025, 1039, 1040, 1055, 1062, 1072, 1097, 1105, 1230
\endgroup 387, 1232
\endinput 1236
\ensuremath 1074
\ERROR *20*, 316, 329, 362, 405, 756, 810
exp commands:
\exp_after:wN 334, 336
\exp_args:No 501
\exp_not:N 491
\exp_stop_f: 330, 339, 341, 343
\expandafter 628, 1018, 1020, 1024, 1025, 1033, 1035, 1039, 1040, 1054, 1062, 1072, 1092, 1093, 1097, 1105, 1118, 1119
\ExplSyntaxOff 401, 587, 993
\ExplSyntaxOn 63, 406, 991

F

\fi 169, 188, 287, 302, 303, 304, 661, 718, 720, 734, 746, 748, 754, 775, 794, 808, 811, 853, 888, 890, 897, 909, 912, 926, 945, 977, 978, 982, 983, 1010, 1014, 1021, 1036, 1046, 1066, 1077, 1084, 1094, 1101, 1111, 1120, 1129
\floatingpenalty 223
fnote commands:
\g_fnote_debug_bool 118, 119, 122, 134
\fnote_footnotemark: 151, 151, 197, 411
\fnote_footnotetext:n 198, 198, 278, 416
\fnote_gput_refs:nn 491, 514, 521, 526, 546
\l_fnote_link_bool *23*, 68, 428, 431, 499
\l_fnote_link_type_tl 70, 503
\fnote_makefntext:n *9*, *19*, *20*, 289, 289, 350
\fnote_mark_gpop:nnN *8*, 465, 465, 478, 543
\fnote_mark_gput:nn *8*, 457, 457, 464, 485
\fnote_new:nn *8*, 451, 451, 456
\fnote_set_fnmark:nn 97, 97, 366
\fnote_step_fnmark:nn 84, 84, 371, 380
\l_fnote_type_tl 66, 485, 543
fnote internal commands:
\l__fnote_autodetect_bool 75, 388, 391, 483, 541
\l__fnote_currentlabel_tl *21*, *25*, 73, 389, 493, 495
\l__fnote_currentrefs_seq 74, 543, 544, 566, 567
\l__fnote_currentstruct_tl *23*, 77, 78, 434, 539, 546, 568, 569
__fnote_debug_footnotemark: 120, 129, 152
__fnote_debug_footnotetext: 132, 148, 199
\l__fnote_dflt_struct_tl 527, 528, 537
__fnote_finish: . . . 179, 179, 195, 374
__fnote_footmark_finish: *15*

__fnote_footnotemark_finish: 177, 179, 195, 372, 374, 376
__fnote_gput_ref:nn 514, 514, 523, 524
__fnote_label_hook: 435, 435, 441, 446
\l__fnote_linktarget_tl 72, 486, 495, 504
__fnote_patch: 318, 353
__fnote_patch_@makefnmark:w 328, 342
__fnote_patch_hb@xt@:w 325, 340
__fnote_patch_hbox:w 322, 338
\l__fnote_patch_seq 361, 362
\l__fnote_patch_tl 315, 320, 321, 324, 327, 333, 334, 336, 361
__fnote_tmp:w 316, 322, 325, 328, 330, 334, 335, 336
\l__fnote_tmpa_prop 64, 468, 469
\l__fnote_tmpa_tl 65
\fnsymbol 633
\footglue 851, 861
\footins . . . *4*, 271, 529, 703, 732, 835, 879
\footnote *7*, *9*, 369, 1016, 1017
\footnotebaselineskip . 842, 864, 871, 878
\footnotelayout . . 663, 666, 667, 958, 979
\footnotemargin *6*, *18*, 280, 281, 285, 286, 292, 293, 295, 298, 301, 646, 690, 693, 918, 923, 924, 925, 941, 942, 970, 971, 973, 976
\footnotemark . . *3*, *4*, *7–9*, 378, 1031, 1032
\footnoterule 702, 707, 708, 710
\footnotesep 22, 221, 273, 656, 887
\footnotesize 14, 214, 870, 952
\footnotetext *5–9*, 364
\footref *8*, *9*, *21*, *23*, 383

G

\gdef 706, 1134
\global . . 12, 395, 732, 745, 871, 903, 1131
\gluestretchorder 740
group commands:
\group_begin: 91, 103
\group_end: 94, 106

H

\hangfootparindent 700, 957
\hangfootparskip 699, 956
\hbox *19*, *20*, 33, 321, 338, 339, 358, 361, 838, 940
hook commands:
\hook_gput_code:nnn 82, 489
\hrule 709
\hsize *5*, *16*, 15, 224, 655, 895, 898
\hskip 851
\hspace 930
\hss 293, 301, 314, 942, 971, 976
\ht 840

I

\IfBooleanTF 1082, 1109
\ifcase 755, 1079, 1106
\ifdim 285, 292, 295, 298, 923, 941, 970, 973, 1002
\ifhbox 883
\ifhmode 159, 186
\ifnum 714, 740, 895, 908
\ifx 711, 737, 1017, 1032, 1062, 1073, 1090, 1097, 1116
\ignorespaces 22, 254
\indent *5*
\input 729
\insert *4*, *15*, 271, 529, 835
\insertpenalties 719
int commands:
\int_compare:nTF 362
\int_set:Nn 92, 104
\interfootnotelinepenalty 220
\interlinepenalty *5*, 220

K

\kern 709, 998, 999

L

\label *8*, *9*
\lastbox 882
\lastkern 1002
\lastskip 739
\leavevmode 158, 954
\leftmargin 947, 950, 951
legacy commands:
\legacy_if:nTF 428
\let 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
\linepenalty 666, 667
\linewidth 949, 950, 951
\llap *6*, 296, 301, 955, 974, 976
\LogHook 124, 125, 126, 127, 128, 142, 143, 144, 145, 146, 147
\long 11, 335, 837, 875, 938, 967

M

\MakeLinkTarget 567, 568
\MakePerPage *35*, 902
\marginpar 653
\marginparwidth 655
\mathparagraph 1148, 1164, 1172, 1183, 1189, 1196, 1204, 1210, 1217
\mathsection . . 1146, 1154, 1163, 1171, 1182, 1188, 1194, 1203, 1209, 1215

\maxdimen *6*, 281, 285, 298, 646, 919
\MessageBreak 1137
\mpfootnotemark 1219
\mpfootnoterule 708, 712
\multfootsep 992, 1008
\multiplefootnotemarker 989, 998, 999, 1002

N

\NeedsTeXFormat 611, 622
\newbox 859, 860
\newcommand 311, 699, 700, 1047, 1219
\newdimen 280, 864
\NewHook . 110, 111, 112, 114, 115, 116, 117
\newif 634, 640, 669, 674, 677, 695, 723, 725, 814, 1060
\NewMirroredHookPair 109, 113
\newskip 861
\newtoks 623
\nobreak *4*, 168, 930
\noexpand 743, 924, 1023, 1024, 1038, 1039, 1074, 1076, 1079, 1081, 1082, 1083, 1084, 1106, 1108, 1109, 1110, 1111
\noindent 309, 313, 559, 969
\normalbaselineskip 871
\normalfont 33
\number 1126

O

\or 757, 776, 795, 1092, 1118

P

\PackageError 642, 649, 1051
\PackageInfo 1063, 1098, 1126
\PackageWarning 1065, 1100
\PackageWarningNoLine 1135
\PackageWarningNoline 924
\pagefootnoterule 707, 715
\par 23, 263, 959
\parindent *5*, 226, 312, 957, 968
\parshape 951
\parskip 956
\patchcmd 402
\penalty 850
prg commands:
\prg_do_nothing: 129, 148, 372
\ProcessOptions 727
prop commands:
\prop_gput:Nnn 81, 459
\prop_gremove:Nn 474
\prop_map_inline:Nn 469
\prop_new:N 64, 453
\prop_set_eq:NN 468
\protect 1017, 1023, 1032, 1038, 1127
\providecommand 421, 422, 612, 613, 624, 989, 992
\ProvidesFile 3
\ProvidesPackage 617

R

\RaggedRight 667
\raggedright 666
\RawNoindent 886
\RawParEnd 880
\ref *21*, 386
ref commands:
\ref_attribute_gset:nnnn 434
\ref_label:nn 79, 448
\ref_value:nn 80, 493, 495
\relax . . . 187, 628, 629, 690, 693, 703, 710, 737, 745, 750, 752, 753, 755, 842, 850, 925, 956, 957, 999, 1002, 1009, 1013, 1062, 1097, 1131, 1230
\renewcommand 633, 636, 666, 667, 702
\RequirePackage 400, 438, 901
\rightmargin 948
\rule 22, 273, 887

S

seq commands:
\seq_clear:N 467
\seq_count:N 362
\seq_map_inline:Nn 544, 567
\seq_new:N 74
\seq_put_right:Nn 473
\seq_set_split:Nnn 361
\seq_show:N 566
\setbox 12, 395, 732, 838, 882, 940
\setfnsymbol 638, 1047, 1198
\setlength 286, 646
\skip 703
\spacefactor 160, 187, 1004, 1009
\splitfootnoterule 709, 717
\splitmaxdepth 222
\splittopskip 221
\stepcounter *35*, 87, 904, 911, 1223
\strut 849
\strutbox 22, 222, 274

T

tag commands:
\tag_check_child:nnTF 532
\tag_get:n 460, 486, 492, 539
\tag_if_active:TF 516
\tag_mc_begin:n . . . 497, 570, 583, 992
\tag_mc_begin_pop:n 511, 554, 562, 579, 992
\tag_mc_end: 509, 572, 585, 992

\tag_mc_end_push: 481, 531, 560, 577, 992
\tag_mc_begin:n 482, 534, 537, 540, 569
\tag_struct_end: . . . 510, 540, 553, 573
\tag_struct_gput:nnn 518
\tag_struct_object_ref:n 518
\tag_struct_use:n 83

TEX and LATEX 2ε commands:

\@@dvance@macro 627, 628
\@@par 953
\@MM 223
\@advance@macro 632
\@arabic 1124
\@auxout 625
\@bsphack 1048, 1125
\@ctrerr 1082, 1109, 1129
\@currentcounter 17, 227
\@currentlabel 18, 228
\@diagnose@fnsymbol@orange 1131, 1132, 1134
\@eha 1052
\@empty 422, 663
\@esphack 1057, 1128
\@finalstrut 22, 274
\@fnsymbol 636, 1054, 1059
\@fnsymbol@orange 1127, 1133
\@footnotemark *3*, *15*, *21*, 197, 373, 381, 390, 411, 1225, 1233
\@footnotetext 3, *16*, 278, 367, 375, 403, 416
\@footnotetext@cfgpoint 4, *9*, 136, 205, 271, 404, 653, 835
\@footnotetext@cfgpointii 4, *5*, 137, 243, 272, 658, 837
\@footnotetext@cfgpointiii 4, 138, 248, 273, 659, 847
\@footnotetext@cfgpointiv 5, 139, 261, 274, 660, 848
\@gobble 1028
\@gobbletwo 421
\@if@bfloats@TF 762, 780, 789
\@if@footnotes@TF 760, 770, 787
\@ifclassloaded 818
\@ifnextchar 1026, 1041, 1220
\@ifpackageloaded 426, 815, 893
\@ifundefined 665, 728, 1049
\@iiiminipage *22*, 402
\@kernel@process@footnotetext *9*, 204, 275, 556
\@kernel@process@makefnmark *9*, 174, 196, 513
\@listdepth 711
\@makecol 408, 413
\@makecol@cfgpoint 759, 768, 778, 786, 797, 803
\@makecol@cfgpointii 731
\@makefnmark *3*, *4*, *6*, *9*, *19*, 33, 175, 293, 296, 299, 301, 314, 327, 342, 343, 942, 944, 971, 974, 976
\@makefntext *3–6*, *9*, *10*, *18–20*, *25*, 21, 226, 240, 272, 320, 335, 348, 350, 412, 658, 838, 938, 967
\@makefntext@cfgpoint 5, *9*, *10*, *27*, 140, 290, 309, 344, 557
\@makefntext@cfgpointii 5, *9*, *10*, *28*, 141, 306, 310, 320, 575
\@makefntext@processX 339, 341, 343, 344
\@mpfn 28, 366, 371, 1223, 1230
\@mpfootins 12, 13, 395, 396
\@mpfootnotetext 8, 11, 403
\@mpfootnotetext@cfgpoint . . 393, 404
\@mplistdepth 711
\@ne 679, 895, 951
\@null 1104
\@outputbox@append 738, 743, 761, 762, 771, 781, 788, 789
\@outputbox@appendfootnotes 764, 773, 779, 792, 798, 805
\@outputbox@attachfloats 765, 769, 783, 791, 799, 804
\@outputbox@reinsertbskip 742, 745, 750, 753, 763, 772, 782, 790, 800, 806
\@outputbox@removebskip 736, 752
\@parboxrestore 16, 225, 877
\@plus 703
\@setpar 953
\@stpelt 903
\@tempa 1089, 1090, 1115, 1116
\@tempb 1068, 1090, 1103, 1116
\@tempboxa 858, 940, 947, 955
\@tempc 1070, 1073
\@tempcnta 628, 629, 630
\@tempd 1071, 1073
\@tempskipa 739, 740, 741, 743
\@textbottom 737
\@textsuperscript 33
\@thefnmark *7–9*, *11*, *13*, *21*, *24*, *25*, *27*, 19, 33, 88, 93, 100, 105, 228, 386, 485, 543, 1224, 1231
\@typeset@protect 1017, 1032
\@x@sf 160, 187, 1004, 1009
\@xmpfootnotemark 1221, 1228
\iiiminipage *10*
\c@footnote 636
\color@begingroup 20, 238
\color@endgroup 24, 264

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

\thempfn 29, 1224, 1231
tl commands:
\tl_if_eq:nnTF 471
\tl_if_in:NnTF 321, 324, 327
\tl_if_novalue:nTF 85, 98
\tl_new:N 65, 66, 70, 72, 73, 77, 315, 527
\tl_set:Nn 67, 71, 78, 320, 333, 389, 486, 495, 528, 539

U

\unhbox 884
\unkern 1006, 1007
\unskip 155
\unvbox 13, 396, 879
use commands:
\use:N 88, 93, 100, 105
\use:n 343
\UseHook 156, 166, 172, 184, 191, 202, 211, 236, 252, 259, 268

V

\value 908
\vbox 12, 395, 732
\vfill 761, 762, 771, 781, 788, 789
\vskip 741, 743

W

\wd 841, 947

X

\xdef 719, 742, 1072, 1105