fnlineno.sty

—

Numbering Footnote Lines*

January 29, 2025

Abstract

fnlineno.sty extends lineno.sty[1] (created by Stephan I. Böttcher) such that even `\footnote` lines are numbered and can be referred to using `\linelabel`, `\ref`, etc.

Making the package was motivated as support for *critical editions* of *printed works with footnotes* as opposed to scholarly critical editions of *manuscripts.* For this purpose, an extension edfnotes of the ednotes package for critical editions, building on fnlineno, is provided by the *ednotes* bundle.[2]

lineno.sty has also been used for the revision process of *submissions.* With fnlineno.sty, reference to footnotes in the submitted work may become possible.

As to *implementation:* 1. Some included tools for *storing and restoring global settings* may be "exported" as standalone packages later. 2. The method of typesetting footnotes on the main vertical list may later lead to applying the line numbering method to several *parallel* texts (with footnotes) and to `inner` material such as table cells.

Keywords: line numbers; footnotes, pagewise, critical editions, revision

Contents

---

*This document describes version [v0.55](#) of fnlineno.sty as of 2011/01/07.
[1] `http://ctan.org/pkg/lineno`
[2] `http://ctan.org/pkg/ednotes`

1   Usage and Features

1.1   Package File Header (Legalize)

```
1   \NeedsTeXFormat{LaTeX2e}[1994/12/01]
2   \ProvidesPackage{fnlineno}[2011/01/07 v0.55
3                             numbers to footnote lines (UL)]
4
5   %% Copyright (C) 2010 Uwe Lück
6   %%
7   %% This file can be redistributed and/or modified under
8   %% the terms of the LaTeX Project Public License; either
9   %% version 1.3c of the License, or any later version.
10  %% The latest version of this license is in
11  %%     http://www.latex-project.org/lppl.txt
12  %% We did our best to help you, but there is NO WARRANTY.
13  %%
14  %% Please report bugs, problems, and suggestions via
```

```
15   %%
16   %%   https://github.com/latex-lineno/lineno
17   %%
```

1.2   Installing and Calling

The file fnlineno.sty is provided ready, installation only requires putting it somewhere where TeX finds it (which may need updating the filename data base).[3]

As usually, fnlineno.sty is loaded by `\usepackage{fnlineno}` below the `\documentclass` line and before `\begin{document}`.

1.3   Limitations

v0.55 should really work the way users expect, but please consider:

1. Nothing is known about compatibility with packages (other than manyfoot and bigfoot) providing footnote features beyond standard LaTeX.

2. `\lipsum[`⟨*opt-arg*⟩`]` in main text produces a different number of paragraphs . . .

3. v0.41 tried supporting `\pagebreak` in footnotes for manual control of splitting footnotes. However, it wrongly assumed that `\pagebreak[4]` forces a footnote split, cf. Section 2.5.3; users better still don't use `\pagebreak` in footnotes!

4. Much of the code is "guessed" without complete knowledge of TeX internals and without having tested many possible cases.

5. *Local* switching to "pagewise" numbering won't be possible for a while; we rather assume that you *always* want "pagewise" numbering.

6. Nothing has been tried to offer choices about the *style* of numbering footnotes.

---

[3]`http://www.tex.ac.uk/cgi-bin/texfaq2html?label=inst-wlcf`

2 Implementation

2.1 Terms

"**OTR**" is short for *"output routine"*, "**MVL**" is short for *"main vertical list"*.

2.2 Basic Strategy

LaTeX's `\@footnotetext` writes the footnote text into the insertion register. For numbering the footnote lines, *we here* do not execute this `\@footnotetext` immediately after placing `\@footnotemark`, but postpone its `\insert` a little so it is executed only after the main text paragraph has been broken into lines. Right below the line that contains the footnote mark, a special new "slot" of the **OTR** is called that interchanges "the page so far" with the footnote text. When the latter has been typeset, another "slot" of the OTR puts "the page so far" back to the MVL and immediately after that fills the footnote text as just typeset on the MVL into the `\insert` register.

Passing footnotes from horizontal mode to vertical mode resembles lineno's `\PostponeVadjust`, but a different **list** `\FNLN@list` must store code (*a*) for the footnote **mark** and (*b*) for the footnote **text**.

2.3 Package Options

A package option `[check-latex]` for checking vital LaTeX internals may once be offered (TODO 2010/12/12) . . .

```
18    \newif\if@FNLN@check@
19    \DeclareOption{check-latex}{\@FNLN@check@true}
20    \ProcessOptions
```

2.4 Footnote Commands

2.4.1 Standard Footnotes

The following macro `\FNLN@ltx@fntext` is a copy of LaTeX's `\@footnotetext` that we are varying. It may be used for a check if the `\@footnotetext` that fnlineno.sty encounters is the one expected (TODO). In line numbering mode, this code may never be needed all at once, rather we will have to see which material must be used at which point of our unusual way of processeing footnotes.

```
21    \if@FNLN@check@
22      \long\def\FNLN@ltx@fntext#1{\insert\footins{%
23        \reset@font\footnotesize
24        \interlinepenalty\interfootnotelinepenalty
25        \splittopskip\footnotesep
26        \splitmaxdepth \dp\strutbox \floatingpenalty \@MM
27        \hsize\columnwidth \@parboxrestore
28        \protected@edef\@currentlabel{%
29          \csname p@footnote\endcsname\@thefnmark
```

```
30          }%
31          \color@begingroup
32            \@makefntext{%
33              \rule\z@\footnotesep\ignorespaces#1\@finalstrut\strutbox}%
34          \color@endgroup}}%
35    \fi
```

2.4.2 Modifying Footnote Commands

In order to number `\footnote` lines and make `\linelabel` available in foot-notes, it seems to suffice (with standard LaTeX) to **redefine** the internal `\@footnotetext`. In line numbering mode, `\@footnotetext` will act as `\FNLN@text`, (*i*) placing a "signal" **output penalty** below the current line via `\vadjust` and (*ii*) appending the footnote text to the **list** `\FNLN@list` of **footnote texts**.

    `\FNLN@@text` stores the `\@footnotetext` found, we might check if it is `\FNLN@ltx@fntext` ...

```
36    \let\FNLN@@text\@footnotetext
37    \def\@footnotetext{%
38        \ifLineNumbers  \expandafter \FNLN@text
39        \else           \expandafter \FNLN@@text
40        \fi}
41    \def \FNLN@text {%                     %% 2010/12/31 arg read later
42        \vadjust{\penalty-\FNLN@M@swap@codepen}%
```

Standard LaTeX's `\@footnotetext` expands `\@thefnmark` to produce the foot-note mark at the page bottom, right after it has been determined for the mark in the main text. *Here* the footnote text will be typeset only when *other* footnote marks may have been formed for typesetting the main text paragraph before. In the **footnote list** macro `\FNLN@list`, the (`&\protect`ed) *current* expansion ⟨*mark*⟩ of `\@thefnmark` is stored as an item preceding the footnote text ⟨*text*⟩. One footnote entry in `\FNLN@list` thus has the form '⟨*mark*⟩`\@lt`⟨*text*⟩`\@lt`'. LaTeX's internal `\g@addto@macro` is used to *append* an entry to the list (at the right). The OTR will later take the entries from the left of the list.

    The argument of the auxiliary/temporary `\@tempa` will contain the footnote text and thus must be able to carry `\par` tokens. We therefore need a `\long` version of `\protected@edef`:

```
43        \let\@@protect\protect
44        \let\protect\@unexpandable@protect
45        \afterassignment\restore@protect
46        \long \edef \@tempa ##1{%
47            \noexpand\g@addto@macro \noexpand\FNLN@list {%
48                \@thefnmark \noexpand\@lt ##1\noexpand \@lt}}%
```

... issuing '`\g@addto@macro\FNLN@list{`⟨*mark*⟩`\elt`⟨*text*⟩`\@lt}`' ...

```
49        \@tempa                         %% reads arg
50    }
```

Here we initialize $\boxed{\texttt{\textbackslash FNLN@list}}$:

```
51    \let\FNLN@list\@empty
```

2.5 Output Routines

2.5.1 lineno's Output Routine

The following is a copy of lineno's OTR that we are varying. It may be used for a check if the OTR that fnlineno.sty encounters is the one expected (TODO).

```
52    \if@FNLN@check@
53      \def\FNLN@lno@output {%
54        \LineNoTest
55        \if@tempswa
56          \ifnum\outputpenalty=-\@Mllbcodepen
57            \WriteLineNo
58          \else
59            \ifnum\outputpenalty=-\@Mppvacodepen
60              \PassVadjustList
61            \else
62              \LineNoLaTeXOutput
63            \fi
64          \fi
65        \else
66          \MakeLineNo
67        \fi
68      }
```

The "signal penalties" used here are

```
69        \mathchardef\FNLN@M@llbl@codepen=11111
70        \mathchardef\FNLN@M@ppva@codepen=11112
71    \fi
```

Their names should mean "`\linelabel` code penalty" and "`\PostponeVadjust` code penalty."

$\boxed{\texttt{\textbackslash TheLineNoLaTeXOutput}}$: It turns out to be inconvenient here that lineno sacrifices access to the *primitive* `\output` ("`\@tempa`"; TODO: auxiliary package before loading lineno!?; later change lineno.sty indeed). So to change the OTR we use `\LineNoLaTeXOutput` as a hook for adding additional cases of `\outputpenalties`. We take a copy of `\LineNoLaTeXOutput` here.

```
72    \let\TheLineNoLaTeXOutput\LineNoLaTeXOutput
```

2.5.2 Tools for Temporary Parameter Changes

$\boxed{\texttt{\textbackslash GStoreReg\{}\langle register\rangle\texttt{\}}}$ (or $\boxed{\texttt{\textbackslash GStoreReg}\langle register\rangle}$

when $\langle register\rangle$ is a single token—'`\count0`' being a counterexample . . .)   stores the current content of $\langle register\rangle$ (*globally*) as an internal macro so that it can

be restored later by

$\boxed{\texttt{\textbackslash RestoreReg\{}\langle\textit{register}\rangle\texttt{\}}}$ (or $\boxed{\texttt{\textbackslash RestoreReg}\langle\textit{register}\rangle}$)

or *globally* by

$\boxed{\texttt{\textbackslash GRestoreReg\{}\langle\textit{register}\rangle\texttt{\}}}$ ($\boxed{\texttt{\textbackslash GRestoreReg}\langle\textit{register}\rangle}$)

(The OTR runs in a local group!—Recall that assignments to "special dimens"—
TeXbook p. 271—are automatically global.) $\langle\textit{register}\rangle$ is something that can be
prefixed by \the to read its content and to which you can assign a value $\langle\textit{value}\rangle$
by '$\langle\textit{register}\rangle\langle\textit{value}\rangle$'. (TODO: could also be some \catcode!)

```
73    \newcommand*{\GStoreReg}[1]{%
74        \expandafter \xdef \csname GS\string#1\endcsname {\the #1}}
75    \newcommand*{\RestoreReg}[1]{#1\csname GS\string#1\endcsname \relax}
76    \newcommand*{\GRestoreReg}{\global\RestoreReg}
```

$\boxed{\texttt{\textbackslash GStoreSetReg\{}\langle\textit{register}\rangle\texttt{\}\{}\langle\textit{value}\rangle\texttt{\}}}$ assigns $\langle\textit{value}\rangle$ to $\langle\textit{register}\rangle$ (locally) af-
ter executing \GStoreSet, $\boxed{\texttt{\textbackslash GStoreGSetReg}}$ does the same *globally* (and still
argument braces aren't needed when a single token refers to the register).

```
77    \newcommand*{\g@storesetreg}[3]{\GStoreReg{#2}#1#2#3\relax}
78    \newcommand*{\GStoreSetReg} {\g@storesetreg\relax}
79    \newcommand*{\GStoreGSetReg}{\g@storesetreg\global}
```

(These preliminaries might go into an own new package, TODO! + loop on list
of $\langle\textit{register}\rangle$s ...)

2.5.3 The basic hook

We use two more penalties triggering the "MVL swaps:"

```
80    \mathchardef\FNLN@M@swap@codepen  =11113
81    \mathchardef\FNLN@M@insert@codepen=11114
```

v0.41 deals with $\boxed{\texttt{\textbackslash pagebreak}}$ in footnote texts, using a flag $\boxed{\texttt{\textbackslash if@FNLN@sw@}}$
that must be set globally. It turned out not to work properly; however, the
new switch has served a different purpose for "continuous line numbering," cf.
section 2.6.

```
82    \newif\if@FNLN@sw@  \global\@FNLN@sw@false  %% v0.41
```

When a \pagebreak triggers the OTR while typesetting the footnote text, the
page content is collected in a box $\boxed{\texttt{\textbackslash FNLN@holdft}}$:

```
83    \newsavebox\FNLN@holdft                    %% v0.41
```

Using $\boxed{\texttt{\textbackslash LineNoLaTeXOutput}}$ for hooking into the OTR:

```
84    \renewcommand*{\LineNoLaTeXOutput}{%
85      \ifnum\outputpenalty=-\FNLN@M@swap@codepen
86        \SwapFootnoteMain
87      \else
```

```
88          \ifnum\outputpenalty=-\FNLN@M@insert@codepen
89            \InsertFootnote
90          \else
91            \if@FNLN@sw@                          %% v0.41
92   %           \showthe\outputpenalty      %% 2010/12/20
93              \global\setbox \FNLN@holdft \vbox{%
94                \unvbox\FNLN@holdft
```

TODO from v0.41: \pagebreak[4] does not seem to force (reliably) splitting a footnote; if the footnote is not split here, at present the \baselineskip is lost, see the footnote paragraph starting with 'C' in edfndemo.pdf as of 2010/12/21. We would need some measuring … \pagebreak might be redefined … resembling LaTeX's \@specialoutput!

```
95                \unvbox\@cclv
```

TODO same problem here, see the footnote paragraph starting with 'D' in edfndemo.pdf as of 2010/12/21.

```
96                \penalty\outputpenalty}%
97              %% TODO reset page book-keeping!?    %% v0.41
98            \else
99              \TheLineNoLaTeXOutput          %% "the real \LineNoLaTeXOuput"
100           \fi
101         \fi
102       \fi
103     }
```

An idea: Instead of so many \ifnum, use

$$\csname\langle chars\rangle\the\outputpenalty\endcsname$$

… in lineno.sty, when you really have a broad range of \outputpenalties useful to be described by \ifnum range checks …

2.5.4 Typesetting the Footnote Text

\SwapFootnoteMain is the slot of the OTR that our modified \@footnotetext calls with \outputpenalty $= -$\FNLN@M@swap@codepen. The "column so far" is stored in a new box register \FLNL@holdcol.

```
104     \newsavebox\FNLN@holdcol
105     \newcommand*{\SwapFootnoteMain}{%
106       \global \setbox\FNLN@holdcol \vbox{\unvbox\@cclv}%
```

(… cf. \@holdpg in LaTeX.)

The entire text of a footnote is typeset on top of the MVL. \vsize is maximized temporarily to avoid that the footnote text is broken across pages.

```
107       \GStoreGSetReg\vsize\maxdimen
```

However, the user may want to use `\pagebreak` in a footnote in order to control manually where a "long" footnote is split. v0.41 tries to support this:

```
108        \global\@FNLN@sw@true                %% v0.41
```

. . . cf. Section 2.5.3.

There shouldn't be any `\topskip`, the space on top of a footnote is controlled by `\footnotesep` entirely:

```
109        \GStoreGSetReg\topskip\z@skip
```

(`\nointerlineskip` as well as setting `\topskip` locally instead fails . . . according to `\showlists` . . .)

Resetting `\pagegoal` (why doesn't it switch to `\vsize = \maxdimen` automatically?), `\pagetotal`, and the other "special dimens" (TEXbook p. 271; rather experimental . . . I think it is important to restore them later . . .)

```
110        \GStoreSetReg\pagegoal \vsize
111        \GStoreSetReg\pagetotal\z@
112        \GStoreSetReg\pagestretch\z@
113        \GStoreSetReg\pagefilstretch\z@
114        \GStoreSetReg\pagefillstretch\z@
115        \GStoreSetReg\pagefilllstretch\z@
116        \GStoreSetReg\pageshrink\z@
117        \GStoreSetReg\pagedepth\z@
```

We must choose certain settings from `\@footnotetext` such as font:

```
118        \reset@font\footnotesize
119        \interlinepenalty\interfootnotelinepenalty
```

LATEX's `split` things here are relevant at `\insert\footins` only: (TODO!?)

```
120   %     \splittopskip\footnotesep
121   %     \splitmaxdepth \dp\strutbox \floatingpenalty \@MM
122        \hsize\columnwidth \@parboxrestore
```

The previous lines were from LATEX's `\@footnotetext`. Now we need to restore the `\@thefnmark` that belongs to the current footnote text. We use a macro that tears two items from `\FNLN@list` and executes the rest of LATEX's `\@footnotetext`:

```
123        \expandafter \FNLN@typeset \FNLN@list \@@
124   %     \showthe\vsize
```

. . . so a `\vsize` assignment without `\global` is noted here, and an analogous `\topskip` assignment is not!? TODO . . .

```
125     }
```

`\FNLN@typeset` first removes something from the list of footnotes, similarly to LaTeX's `\@xnext` and lineno's `\@LN@xnext`, then executes a remaining portion of LaTeX's `\@footnotetext`. The footnote text may contain `\par` tokens, so the definition must be `\long`:

```
126    \long\def \FNLN@typeset #1\@lt #2\@lt #3\@@{%
127        \gdef\FNLN@list{#3}%
128        \def\@thefnmark{#1}%
```

This was our own, and next LaTeX continues:

```
129        \protected@edef\@currentlabel{%
130          \csname p@footnote\endcsname\@thefnmark
131        }%
132        \color@begingroup
```

We insert starting the lineno settings . . .

```
133          \linenumbers
134          \setfootnotelinenumbers              %% 2010/12/25
```

. . . LaTeX again (v0.41 exports dealing with closing `\par` to `finstrut.sty`):

```
135          \@makefntext{%
136            \rule\z@\footnotesep\ignorespaces
```

We replace #1 by #2\par (`\linenumberpar`), so we really need `finstrut.sty`:

```
137            #2\par
138            \@finalstrut\strutbox}%
139          \color@endgroup
```

Now we trigger the "swap back slot" of the OTR:

```
140          \penalty-\FNLN@M@insert@codepen
141    }
142    \RequirePackage{finstrut}
```

2.5.5 \insert the Footnote Text

`\InsertFootnote` is the slot of the OTR that executes `\insert\footins` with the numbered footnote text. The "column so far" stored in `\FNLN@holdcol` is put onto the top of the MVL, and then parts of LaTeX's `\@footnotetext` are performed that haven't been done earlier, applied to the footnote text that the OTR should have found in `\box255`. Before however, the previous `\topskip`, `\vsize`, and the `\page...` book-keeping parameters are restored:

```
143    \newcommand*{\InsertFootnote}{%
144        \GRestoreReg\topskip    \GRestoreReg\vsize
```

($\ldots$ *global* restoring of `\vsize` proved vital with edfndemo 2010/12/17 $\ldots$)

```
145        \RestoreReg \pagegoal      \RestoreReg\pagetotal
146        \RestoreReg \pagestretch
147        \RestoreReg \pagefilstretch
148        \RestoreReg \pagefillstretch
149        \RestoreReg \pagefilllstretch
150        \RestoreReg \pageshrink    \RestoreReg\pagedepth
151        \unvbox\FNLN@holdcol
152        \insert\footins{%
153            \splittopskip\footnotesep
154            \splitmaxdepth \dp\strutbox \floatingpenalty \@MM
```

Support of `\pagebreak` with v0.41:

```
155            \unvbox\FNLN@holdft              %% v0.41
156            \unvbox\@cclv}%
157        \global\@FNLN@sw@false              %% v0.41
```

With v0.5, global settings for "pagewise" numbering must be restored:

```
158        \unsetfootnotelinenumbers
159    }
```

2.6   "Continuous" Numbering

2.6.1   Goal

With v0.5, for the first time we try to get a "pagewise" numbering such that, if a main text line has a footnote, (*i*) its printed number is just the natural successor of the printed number of the previous main text line (instead of continuing previous numbering with the lines of the footnote first), and (*ii*) the printed numbers of footnote lines just continue the printed numbers of the main text lines. This "obvious" desirement is not easy to achieve; already pagewise numbering of main text lines, without numbering footnote lines, has been somewhat ingenious.

2.6.2   How to Number Lines Pagewise

The basic idea of lineno's pagewise numbering is:

1. Each numbered line of the document is identified by a unique counter value, an "absolute" number.

2. For each page (and column), the range of absolute line numbers occurring on them is recorded (or actually: the first and the last number).

3. The "public," "human-readable" ("pagewise") format of a given absolute line number l is generated by (*i*) finding the page (and column) with first number n and last number k such that $n \leq l \leq k$, (*ii*) "printing" $l - n + 1$ in "columnwise" mode, otherwise $l - m + 1$ where m is the first absolute line number in the left-hand column of the same page.

Generating the "pagewise" representation for a given absolute line number l thus may be summarized as *finding the corresponding* **offset** *value* to be subtracted $(n, n+1, m, \text{or } m+1 \dots)$.

When *footnote* lines are to be numbered as well, a little problem is the order in which main text and footnote lines increment the absolute counter. lineno's mechanism for this is started immediately after a paragraph has been broken into lines. Each line of the paragraph then calls a macro generating the line number. fnlineno now interrupts numbering of main text lines at a line issuing a footnote. The footnote text is typeset, including numbering its lines at each end of a footnote paragraph. When the footnote text has been sent into the \insert register, numbering of main text lines is resumed.

Up to v0.4 (a development version), we used the *same* absolute counter for main text and footnote lines. When a page p has more than one main text line and the first one has a long footnote continued on the next page $p+1$, there is no "range" of absolute line numbers characterizing page p any more, because the greatest absolute line number of page p exceeds the absolute line numbers of the footnote continued on page $p+1$.

lineno's procedure can be revived by numbering main text lines and footnote lines independently from each other. We use *two* absolute counters, one is incremented with main text lines only, the other with footnote lines only. Numbering of main text lines just will not be affected by numbering of the footnote lines.

Almost the same will hold for footnote lines. Each page (and column) will have a characteristic "range" of absolute footnote line numbers $\{n, \dots, k\}$. The only notable difference will be that for footnote line l we print $(l-n+1)+(K-N+1) = (K+l)-(N+n)+2$ instead of $l-n+1$—where $\{N, \dots, K\}$ is the range of *main text* line numbers of the page (and column).

The previous discussion of **generating** the printed line number from its absolute version has assumed that corresponding **offset** values have been given somehow, or that the "line number ranges" for pages are known from somewhere. In fact, these ranges are **computed** at the **start** of a LaTeX run *before* typesetting, when reading the .aux file for the first time. They are used in the entire document. While typesetting, each numbered line of main text leaves a record of its absolute number and page number in the new version of the .aux file that the run creates, a two-parameter macro \@LN. With fnlineno.sty, there will be new \@FLN entries of the same type. These .aux entries are used for building the page range data for the next run. When the document source has been changed, at least *two runs* will usually be required to get correct line numbers in page margins, and *another* run will be needed so references to line numbers by \ref and \linelabel are correct.

2.6.3 Summary of Changes

Variants of lineno.sty's code for "pagewise" numbering are following. Sometimes we generalize `pagewise` stuff from lineno and re-implement pagewise numbering of main text lines as a special case, the other special case being numbering of footnote lines.

Five things need modifications:

Building page info macros: Processing `\@LN` and `\@FLN` `.aux` entries will use shared building macros, the difference is obtained by switching **name spaces**. (It may be notable that a page may get one info macro for main text and another for footnote text, if it contains footnote text.)

Logging: While typesetting, the shared logging macro is switched to write either `\@LN` or `\@FLN` to the `.aux` file. Also, `\c@linenumber` may refer to either the main text or to the footnote text counter.

Generating "pagewise" format: The choice of `\c@linenumber` also determines which counter is incremented, and again name spaces for page info macros are switched. For footnote lines, a tail macro for adding the number of main text lines will be activated.

Referencing: The `.aux` file may have entries from `\linelabel` containing large numbers from an "absolute" counter. In generating the "human-readable" number, it must be known whether it is a main text or a footnote line number. An additional complication is referring to a main text line from a footnote and vice versa—thinking of global changes in generating the number. Or even think of the case referring from unnumbered text to numbered text! (I have wondered before if the entry couldn't be the ready human-readable number, TODO!)

Lists of "vertical tasks": lineno.sty (v4) has introduced two lists of tasks that were issued in horizontal mode but only can be completed after breaking a paragraph into lines: one for `\linelabel`s and one for `\vadjust` items that must wait until the line number has been attached. It is essential that the tasks are processed in the same order in vertical mode as they were issued in horizontal mode. As we are now interrupting processing of main text paragraphs for processing footnotes, tasks for footnote text must be lined up in separate lists than tasks for main text. This is indeed essential for the previous issue of getting `\linelabel` work in footnotes as well as in main text.

2.6.4 Info Building

`\@LN`, `\@FLN`, and `\@FNLN` are processed at reading the `.aux` file before typesetting only. The **interface** to **generating** "pagewise" and footnote line numbers just are `\LN@Pfirst` and `\FLN@Pfirst`, eventually pointing to the first page/column with numbered main text lines or footnote lines, resp.

160   `\def \FLN@Pfirst {\nextLN\relax}`

This initialization of `\FLN@Pfirst` is just the same as the one of `\LN@Pfirst` in lineno.sty; their expansions are changed as soon as such a page is found, replacing the `\relax` by the corresponding page info macro.

`\LN@Pfirst` and `\FLN@Pfirst` are passed to `\testFirstNumberedPage` via the hook `\FNLN@first@numbered` that by default is the same as `\LN@first`:

```
161    \def \FNLN@first@numbered {\LN@Pfirst}
```

(oh, it must be `\def` here to recognize the change . . .). This must be changed by `\setfootnotelinenumbers` (`\let` then, as when called the change will have happened).

Moreover, they are passed to `\NumberedPageCache` (the page info macro where a search starts, "current" page/column) as its initialization; the "generating" macros then change the latter macro following `\nextLN` in the page info macros.

In this sense, no other "name space switching" is needed for communication with other functions.

lineno.sty has changed `\LastNumberedPage` globally . . . the last page with numbered *footnote* lines may well be another one than the last page with numbered *main* text lines . . . But fortunately, also `\LastNumberedPage` is needed in reading the `.aux` before typesetting only (`\@onlypreamble` is LaTeX's disabling command):

```
162    \@onlypreamble\LastNumberedPage
```

In lineno.sty, we have `\def\LastNumberedPage{first}`. We need the same for the footnote variant `\FNLN@last@numbered` (to be handled globally!):

```
163    \global   \let \FNLN@last@numbered \LastNumberedPage
164    \@onlypreamble \FNLN@last@numbered
```

`\@FNLN{⟨names⟩}⟨last-numbered⟩{⟨line⟩}{⟨page⟩}`

generalizes lineno.sty's `\@LN{⟨line⟩}{⟨page⟩}` to re-implement it. There is an additional parameter argument ⟨*names*⟩ for choosing name spaces and a parameter ⟨*last-numbered*⟩ for choosing the macro storing the "last numbered page." (An argument without braces expects a macro name.)

```
165    \newcommand* \@FNLN [4]{{%
166        \expandafter\@@LN
167            \csname #1#4C\@LN@column \expandafter\endcsname
168            \csname #1O#4\endcsname
169            {#3}{#4}{#1}{#2}}}
170    \@onlypreamble\@FNLN
```

As in lineno.sty `\@LN` calls `\@@LN`, a new variant of `\@@LN` is called by `\@FLN` here, but it gets one additional parameter for passing ⟨*names*⟩ and another for passing ⟨*last-numbered*⟩ from `\@FLN`. So the new syntax is

`\@@LN⟨info⟩⟨first-page-line⟩{⟨line⟩}{⟨page⟩}{⟨names⟩}⟨last-numbered⟩`:

```
171    \renewcommand* \@@LN [6]{%
172      \ifx#1\relax
173        \ifx#2\relax\gdef#2{#3}\fi
174        \expandafter\@@@LN\csname #5#6\endcsname#1%
```

```
175        \xdef#1{\lastLN{#3}\firstLN{#3}%
176               \pageLN{#4}{\@LN@column}{#2}\nextLN\relax}%
177      \else
178        \def\lastLN##1{\noexpand\lastLN{#3}}%
179        \xdef#1{#1}%
180      \fi
181      \xdef#6{#4C\@LN@column}}
182    \@onlypreamble\@@LN
```

lineno.sty's $\boxed{\texttt{\textbackslash@@@LN}}$ does not need any adjustment.

lineno.sty's $\boxed{\texttt{\textbackslash@LN\{}\langle line \rangle\texttt{\}\{}\langle page \rangle\texttt{\}}}$ is reimplemented as

```
183    \def \@LN {\@FNLN{LN@P}\LastNumberedPage}
```

—so `\@LN` really does the same as before, including name spaces. $\boxed{\texttt{\textbackslash@FLN\{}\langle line \rangle\texttt{\}\{}\langle page \rangle\texttt{\}}}$ is the other special case of the new `\@FNLN`—an F precedes the earlier names, and $\boxed{\texttt{\textbackslash FNLN@last@numbered}}$ is the storing macro initialized above:

```
184    \def \@FLN {\@FNLN{FLN@P}\FNLN@last@numbered}
```

For logging, we make both unexpandable:

```
185    % \AtBeginDocument{\let\@LN\relax \let\@FLN\relax}
```

... but this way nothing appears in the file!? TODO ...

```
186    \@onlypreamble\@LN   \@onlypreamble\@FLN
```

For reading the `.aux` finally, we do what lineno does with `\@LN`:

```
187    \AtEndDocument{\let\@FLN\@gobbletwo}
```

2.6.5 Tool for Reusing Global Operations with Macros

lineno.sty v4 provides list handling (changing lists globally) and global changes of `\NumberedPageCache`. We want to use them in "main text" mode as well as in "footnote" mode. To use such an operation on $\langle ln\text{-}macro \rangle$ for $\langle fln\text{-}macro \rangle$, we `\global\let`$\langle ln\text{-}macro \rangle \langle fln\text{-}macro \rangle$, apply the operations, and finally `\global\let`$\langle fln\text{-}macro \rangle \langle ln\text{-}macro \rangle$. However, we are not only interested in how $\langle fln\text{-}macro \rangle$ is changed this way, rather $\langle ln\text{-}macro \rangle$ also is used as input for some operations, and we can choose which $\langle fln\text{-}macro \rangle$ should be used as input. To switch from working on/with $\langle fln\text{-}1 \rangle$ to $\langle fln\text{-}2 \rangle$ using $\langle ln\text{-}macro \rangle$ with an option to use $\langle fln\text{-}1 \rangle$ later again, a tool $\boxed{\texttt{\textbackslash GStoreUse}\langle ln\text{-}macro \rangle \langle fln\text{-}1 \rangle \langle fln\text{-}2 \rangle}$ is provided (should render later switchings much better readable):

```
188    \newcommand* \GStoreUse [3]{\global\let#2#1\global\let#1#3}
```

I.e., current content of `#1` is stored in `#2`, then `#1` attains the content of `#3`.

2.6.6 General Settings for Typesetting Stage

Oh my dear, it seems that all the switching for the footnote variant of `pagewise` must be global (I can't find something useful using `\aftergroup` quickly). Therefore, I render lineno's `\setpagewisenumbers` acting globally:

```
189    \renewcommand*\setpagewiselinenumbers{%
190        \global\let \theLineNumber  \thePagewiseLineNumber
191        \global\let \c@linenumber   \c@pagewiselinenumber
192        \global\let \makeLineNumber \makePagewiseLineNumber
193    }
```

I just force this, hehe . . .

```
194    \setpagewiselinenumbers
```

As a counterpart to `\c@pagelinenumber`, `\c@footnotelinenumber` is reserved for the absolute footnote line numbers:

```
195    \newcount\c@footnotelinenumber
```

`\FNLN@@cache` stores `\NumberedPageCache` as from "main" mode:

```
196    \let \FNLN@@cache \NumberedPageCache
```

`\FNLN@cache` stores `\NumberedPageCache` as from "footnote" mode; its initial content is the counterpart or analogue to `\LN@Pfirst`:

```
197    \def \FNLN@cache {\FLN@Pfirst}
```

`\FNLN@foot@cache` and `\FNLN@main@cache` switch `\NumberedPageCache`:

```
198    \def \FNLN@foot@cache {%
199        \GStoreUse \NumberedPageCache \FNLN@@cache \FNLN@cache}
200    \def \FNLN@main@cache {%
201        \GStoreUse \NumberedPageCache \FNLN@cache \FNLN@@cache}
```

`\FNLN@labels` will be the counterpart to lineno.sty's `\@LN@labellist`:

```
202    \global\let \FNLN@labels \@empty
```

`\FNLN@vadjusts` will be the counterpart to lineno's `\@LN@vadjustlist`:

```
203    \global\let \FNLN@vadjusts \@empty
```

Settings for footnote line numbers first resemble `\setpagewiselinenumbers`; but more changes are needed, and results from main text numbering must be stored. Some of the settings are needed *locally* for generating numbers for labels, collected in `\setgetfootnotelinenumbers`; for this purpose nothing must be stored explicitly:

```
204    \newcommand* \setgetfootnotelinenumbers {%
```

Change of `\theLineNumber` is omitted as we are *reading*, not writing a label.

```
205        \let\c@linenumber\c@footnotelinenumber
206    %        \let\makeLineNumber\makeFootnoteLineNumber
```

But in fact, `\makeFootnoteLineNumber` and `\makePagewiseLineNumber` will be the same. The difference is made by the choice of `\FNLN@first@numbered` and `\NumberedPageCache` for the line range searches.

```
207        \let \FNLN@first@numbered \FLN@Pfirst
208        \let \FNLN@finish \FNLN@add
209    }
```

`\setfootnotelinenumbers` performs all the settings for typesetting footnotes in line numbering mode *globally*, including storing results from typesetting main text:

```
210    \newcommand* \setfootnotelinenumbers {%
211        \globaldefs\@ne
```

The previous line also renders `\setgetfootnotelinenumbers` global:

```
212        \setgetfootnotelinenumbers
```

`\theLineNumber` is used for `\linelabel` entries. `\thePagewiseLineNumber` is replaced by `\theFootnoteLineNumber`:

```
213        \let\theLineNumber\theFootnoteLineNumber
```

Logging to `.aux`:

```
214        \def \FNLN@log {\string\@FLN}%
```

Starting range search: `\NumberedPageCache`

```
215        \FNLN@foot@cache
```

Reusing lineno's task list operations:

```
216        \GStoreUse \@LN@labellist \FNLN@@labels \FNLN@labels
217        \GStoreUse \@LN@vadjustlist \FNLN@@vadjusts \FNLN@vadjusts
218        \globaldefs\z@
219    }
```

For switching back to "main text mode," again some settings may need a local variant—for processing line references from footnotes to main text! This is the purpose of `\setgetpagewiselinenumbers`:

```
220    \newcommand* \setgetpagewiselinenumbers {%
221        \let \FNLN@first@numbered \LN@Pfirst
222        \let \FNLN@finish         \@gobbletwo
223    }
```

`\unsetfootnotelinenumbers` stores the "current" page with footnote lines and loads the "most recent" page with main text lines—and more . . . :

```
224    \newcommand* \unsetfootnotelinenumbers {%
225        \gdef \FNLN@log {\string\@LN}%
226        \FNLN@main@cache
```

Task lists:

```
227        \GStoreUse \@LN@labellist \FNLN@labels \FNLN@@labels
228        \GStoreUse \@LN@vadjustlist \FNLN@vadjusts \FNLN@@vadjusts
229        \globaldefs\@ne \setgetpagewiselinenumbers \globaldefs\z@ %% v0.53
230        \setpagewiselinenumbers
231    }
```

`\makeFootnoteLineNumber` actually only copies `\makePagewiseLineNumber`, different results are obtained be changing hooks. The command first calls logging—`\logtheLineNumber`, then generating the "public" line number—`\getLineNumber` (which in turn only is a copy of `\testNumberedPage` in lineno.sty).

```
232    \@ifdefinable\makeFootnoteLineNumber
233        {\let \makeFootnoteLineNumber \makePagewiseLineNumber}
```

2.6.7   Logging

`\logtheLineNumber` is redefined to log both main text and footnote line numbers.

```
234    \def \logtheLineNumber {%
235        \protected@write\@auxout{}{%
236            \FNLN@log{\the\c@linenumber}{\noexpand\the\c@LN@truepage}}}
```

`\FNLN@log` is the hook for the difference, its default expansion `\@LN` is made for *main text* line numbers:

```
237    \gdef \FNLN@log {\string\@LN}
```

2.6.8   "Public" Line Numbers

Fortunately, these commands don't need to know much about name spaces. The interfaces to them are `\NumberedPageCache`—changing globally—and `\FNLN@first@numbered`. Our `\FNLN@cache` is initialized by analogy to its counterpart `\NumberedPageCache` (a minute name space change):

```
238    \def \FNLN@cache {\FLN@Pfirst}
```

`\testFirstNumberedPage{⟨integer⟩}` from lineno.sty is modified by replacing
`\LN@Pfirst` only:

```
239    \renewcommand* \testFirstNumberedPage [1]{%
240      \ifnum#1>\c@linenumber
241        \def\nextLN##1{%
242          \testNextNumberedPage\FNLN@first@numbered}%
243      \else
244        \let\nextLN\@gobble
245        \def\pageLN{\gotNumberedPage{#1}}%
246      \fi}
```

`\testNumberedPage` and `\testNextNumberedPage` from lineno don't need any
modification. `\testLastNumberedPage` is modified in edfnotes.sty.
    `\gotNumberedPage` just needs a closing hook `\FNLN@finish` to allow for
footnote lines.

```
247    \renewcommand* \gotNumberedPage [4]{%
248      \oddNumberedPagefalse
249      \ifodd \if@twocolumn #3\else #2\fi\relax\oddNumberedPagetrue\fi
250      \advance\c@linenumber\@ne
251      \ifcolumnwiselinenumbers
252        \subtractlinenumberoffset{#1}%
253      \else
254        \subtractlinenumberoffset{#4}%
255      \fi
256  %  \show\FNLN@finish
257      \FNLN@finish{#2}{#3}%
258    }
```

`\FNLN@finish{⟨page⟩}{⟨column⟩}` gobbles both arguments with *main* text
lines, but will add the number of main text lines to *footnote* line numbers:

```
259    \global\let \FNLN@finish \@gobbletwo
```

Then it will act as `\FNLN@add`. We run the page info macro for the same page
(column; if defined).

```
260    \newcommand* \FNLN@add [2]{%
261      \expandafter \let\expandafter \@tempa\csname LN@P#1C#2\endcsname
262      \ifx\@tempa\relax
263      \else
264        \advance\c@linenumber\@ne
265        \ifcolumnwiselinenumbers
266          \let\firstLN\subtractlinenumberoffset
```

. . . rather assuming `\realpagewiselinenumbers`.

```
267          \let\pageLN\@gobblethree
268        \else
269          \let\firstLN\@gobble
270          \def\pageLN##1##2##3{\subtractlinenumberoffset{##3}}%
```

```
271        \fi
272        \def\lastLN##1{\subtractlinenumberoffset{-##1}}%
273        \let\nextLN\@gobble
```

... TODO all needed?

```
274        \@tempa
275      \fi
276    }
```

2.6.9 Referencing

Now that we are using two separate counters for main text lines and foot-note lines (v0.5), correct references to footnote lines using $\boxed{\texttt{\textbackslash linelabel}}$ and \ref need further adjustments.   lineno.sty's \thePagewiseLineNumber and \getpagewiselinenumber{⟨*integer*⟩} are generalized and re-implemented by macros that then serve to implement referring to footnote line numbers.

$\boxed{\texttt{\textbackslash theWiseLineNumber\{⟨\textit{trans}⟩\}}}$ leaves a \protected call to a one-parameter macro ⟨*trans*⟩ in the .aux file:

```
277    \newcommand* \theWiseLineNumber [1]{\protect #1{\the\c@linenumber}}
```

$\boxed{\texttt{\textbackslash getwiselinenumber\{⟨\textit{choice}⟩\}\{⟨\textit{integer}⟩\}}}$ executes ⟨*choice*⟩ before applying \testNumberedPage to ⟨*integer*⟩—within a local group:

```
278    \newcommand* \getwiselinenumber [2]{{%
```

Some wisdom is needed to take account of the current "numbering state" from which \ref was called.

Referring to main text line:

- Unless called from numbered footnote, no extra care is needed.
- If called from numbered footnote, \setgetpagewiselinenumbers and temporary switching of \NumberedPageCache is needed.

Referring to footnote line:

- If called from numbered footnote, no extra care is needed.
- Otherwise, \setgetfootnotelinenumbers and temporary switching of \NumberedPageCache is needed.

```
279      \ifx#1\relax          %% to main text
280        \if@FNLN@sw@         %% from footnote
281          \setgetpagewiselinenumbers
282          \FNLN@main@cache
283          \let \FNLN@restore@cache \FNLN@foot@cache
284        \fi
285      \else                 %% to footnote
286        \if@FNLN@sw@ \else  %% from elsewhere
287          #1%
```

```
288        \FNLN@foot@cache
289        \let \FNLN@restore@cache \FNLN@main@cache
290      \fi
291    \fi
292    \c@linenumber #2\relax\testNumberedPage
293    \thelinenumber
294    \FNLN@restore@cache
295  }}
296  \let \FNLN@restore@cache \relax
```

$\boxed{\texttt{\textbackslash getpagewiselinenumber}}$ doesn't need any ⟨*choice*⟩—we assume that the label was written in the default `pagewise` mode (but it is difficult, though, `\relax` is essential!):

```
297    % \renewcommand* \getpagewiselinenumber {\getwiselinenumber\relax} %!!
```

2010/12/31, a compatibility problem with ednotes' `\newlabel` mechanism shows up. ednotes "undefines" `\getpagewiselinenumber` and restores it only `\AtBeginDocument`. We must ensure that ednotes will not override our new version of `\getpagewiselinenumber`. (TODO in my view another motivation to write "ready" numbers without `\getpagewiselinenumbers` directly.)

We might assume that ednotes (if at all) is loaded directly and loads lineno.sty (that is the usual and recommended way of using ednotes) and that this will happen before fnlineno.sty is loaded. But now that we have spent some time understanding the situation, we can deal with the case as well that lineno.sty is loaded first, then fnlineno.sty is loaded, and then ednotes. (I have assumed earlier that fnlineno.sty is loaded after lineno.sty . . .)

```
298    \AtBeginDocument{%
299        \def \getpagewiselinenumber {\getwiselinenumber\relax}% sic!
300        \let \@EN@getpagewiselno \getpagewiselinenumber}
```

For $\boxed{\texttt{\textbackslash thePagewiseLineNumber}}$, ⟨*trans*⟩ is `\getpagewiselinenumber`:

```
301    \renewcommand* \thePagewiseLineNumber {%
302        \theWiseLineNumber\getpagewiselinenumber}
```

$\boxed{\texttt{\textbackslash getfootnotelinenumber\{⟨integer⟩\}}}$ considers ⟨*integer*⟩ the absolute number of a *footnote* line. The ⟨*choice*⟩ therefore is `\setgetfootnotelinenumbers`:

```
303    \newcommand* \getfootnotelinenumber {%
304        \getwiselinenumber\setgetfootnotelinenumbers}
```

Finally, $\boxed{\texttt{\textbackslash theFootnoteLineNumber}}$ is how `\linelabel` refers to a *footnote* line. `\theWiseLineNumber` is called with ⟨*trans*⟩ being `\getfootnotelinenumber`:

```
305    \newcommand* \theFootnoteLineNumber {%
306        \theWiseLineNumber\getfootnotelinenumber}
```

2.7 Leaving the Package File

```
307    \endinput
```

3 Acknowledgements

On the texhax mailing list, Boris Veytsman recommended using Victor Eijkhout's *TEX by Topic* to me, and Andrej Lapshin pointed me to David Salomon's work on output routines (TUGboat 1990 and 1994, also available as a book, as Ulrich Dirr tells me). It helped me a lot to read about output routines in these works, beyond the TEXbook. The abbreviations 'OTR' and 'MVL' are Salomon's.—And recall Christian's work and support by the DFG named at the start of the package file.—And . . . the ideas of how to implement (*i*) attaching line numbers, (*ii*) \linelabel, and (*iii*) numbering lines "pagewise"—so flexibly, compatibly with many other LATEX packages, still are Stephan's . . .

4 VERSION HISTORY

```
308   v0.1    2010/12/08  very first, \linelabel works in footnote
309           SENT TO Christian, problems with "long" footnotes
310
311   v0.2    2010/12/08  corr. "manifoot"
312           2010/12/09  moving doc. from .tex to here,
313                       different doc. sectioning;
314                       \@footnotetext modified (user feature!);
315                       \@doclearpage NOT modified!; \if@FNLN@placing@
316           2010/12/10  ignore dummy footnote split;
317                       \FNLNpar, \AutoPars, \ExplicitPars,
318                       more on limitations
319           2010/12/11  more trying, almost anew ...
320           JUST STORED
321
322   v0.3    2010/12/12  new approach, removed much before proceeding
323           2010/12/13  -- this was putting \box\footins onto MVL,
324                       bad with those penalties
325           JUST STORED
326
327   v0.4    2010/12/14  another new approach:
328                       typeset footnote on MVL immediately --
329                       described strategy
330           2010/12/15  ... continued, choice of hooking into \output
331                       (...swap...)
332           2010/12/16  ... continued; rearranged sections ...
333                       \FNLN@@fntext vs. ...ltx...
334           2010/12/17  success with \pagegoal ...; \GStoreReg etc.;
335                       ...@fntext shortened
336           2010/12/18  another two limitations: \pagebreak in fn.,
337                       guessed/tested; another note to <register>;
338                       ack. Christian; directed -> organized!?
339           SENT TO Christian/Stephan
340
341   v0.41   2010/12/19  support of \pagebreak with \if@FNLN@sw@ etc.;
```

```
342                      TODO on lists of <register>s
343           2010/12/20  debugging: \if...true; \setbox...ft;
344                       \@finalstrut in vmode exported to finstrut.sty;
345                       notes on how v0.41 still fails with \pagebreak
346           2010/12/21  additional notes on *two* \pagebreak's
347
348   v0.5    2010/12/21  restructuring doc., check@latex@ -> check@,
349                       own account of lineno's pagewise mode
350           2010/12/22  ... continued ...
351           2010/12/23  ... continued ...
352           2010/12/24  ... continued ...
353           2010/12/25  moved this to pwlineno, replaced ...
354                       more on \FNLN@typeset, + \setfootnotelinenumbers
355           2010/12/26  new summary of implementation,
356                       rearranged code sections; logging settled
357   v0.51   2010/12/27  "build" settled, typesetting, logging reformated;
358                       ack.s: "recall"; all settings global,
359                       "public" works
360           JUST STORED, MARGINAL NUMBERS OK,
361           \linelabel in footnote broken
362           [2010/12/28]
363   v0.52   2010/12/28  own label and vadjust lists for footnotes;
364                       local settings for referencing,
365                       tool and care for global changes (...Cache)
366                       (TODO write ready in .aux? needs another run)
367           \linelabel's ok, MARGINAL NOTES MAIN BROKEN
368   v0.53   2010/12/28  debugging; OK; minor doc. modifications;
369                       less "limitations"; \\[\smallskipamount]
370           TO CHRISTIAN 2010-12-29
371   v0.54   2010/12/31  typo options; \FNLN@text without arg,
372                       \getpagewiselinenumber with ednotes
373           2011/01/01  \FNLN@cache, \FNLN@@cache initialized;
374                       doc. "Typesetting Stage" qualification
375           2011/01/02  that qualification was wrong
376           2011/01/03  samepage@hook
377           TO CHRISTIAN SAME DAY
378   v0.55   2011/01/04  samepage@hook emptied here as well;
379           2011/01/06  edited version history
380           2011/01/07  note on \if@FNLN@sw@ with v0.5;
381                       finally without support for samepage@hook!
382                       note on \testLastNumberedPage
383           PART OF EDFN RELEASE r0.5 (together with edfnotes v0.2)
384   v0.55a  2011/02/09  corr. owner; "Limitations" updated; \pagebreak
385
```