fnlineno.sty

Numbering Footnote Lines*

Uwe Lück†

May 20, 2023

Abstract

fnlineno.sty extends lineno.sty1 (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

1 Usage and Features 2

1.1 Package File Header (Legalize) ....................... 2
1.2 Installing and Calling .................................. 3
1.3 Limitations ............................................. 3

*This document describes version v0.55 of fnlineno.sty as of 2011/01/07.
†http://contact-ednotes.sty.de.vu
‡http://ctan.org/pkg/lineno
‡http://ctan.org/pkg/ednotes
1 Usage and Features

1.1 Package File Header (Legalize)

\NeedsTeXFormat{LaTeX2e}[1994/12/01]
\ProvidesPackage{fnlineno}[2011/01/07 v0.55
\quad numbers to footnote lines (UL)]

%% Copyright (C) 2010 Uwe Lück
%%
%% This file can be redistributed and/or modified under
%% the terms of the LaTeX Project Public License; either
%% version 1.3c of the License, or any later version.
%% The latest version of this license is in
%% \url{http://www.latex-project.org/lppl.txt}
%% We did our best to help you, but there is NO WARRANTY.
%%
%% Please report bugs, problems, and suggestions via

2 Implementation

2.1 Terms ............................................. 4
2.2 Basic Strategy ..................................... 4
2.3 Package Options ................................... 4
2.4 Footnote Commands ................................ 4
   2.4.1 Standard Footnotes ........................... 4
   2.4.2 Modifying Footnote Commands ................. 5
2.5 Output Routines .................................. 6
   2.5.1 \texttt{\insert}‘s Output Routine .......... 6
   2.5.2 Tools for Temporary Parameter Changes ....... 6
   2.5.3 The basic hook ................................ 7
   2.5.4 Typesetting the Footnote Text ............... 8
   2.5.5 \texttt{\insert} the Footnote Text .......... 10
2.6 “Continuous” Numbering ............................ 11
   2.6.1 Goal ........................................ 11
   2.6.2 How to Number Lines Pagewise ................. 11
   2.6.3 Summary of Changes ........................... 12
   2.6.4 Info Building ................................ 13
   2.6.5 Tool for Reusing Global Operations with Macros
               ......................................... 15
   2.6.6 General Settings for Typesetting Stage ....... 16
   2.6.7 Logging ...................................... 18
   2.6.8 “Public” Line Numbers ......................... 18
   2.6.9 Referencing ................................... 20
2.7 Leaving the Package File .......................... 21

3 Acknowledgements

4 Version History

1 Usage and Features
1 USAGE AND FEATURES

This work has been supported by the Deutsche Forschungsgemeinschaft (DFG), organized by Prof. Dr. Dr. Christian Tapp at Ruhr-Universität Bochum, Germany. Christian also has constructed some critical tests.

1.2 Installing and Calling

The file \texttt{fnlineno.sty} is provided ready, installation only requires putting it somewhere where \LaTeX{} finds it (which may need updating the filename data base).\footnote{\url{http://www.tex.ac.uk/cgi-bin/texfaq2html?label=inst-wlcf}}

As usually, \texttt{fnlineno.sty} is loaded by \texttt{\usepackage{fnlineno}} below the \texttt{\documentclass} line and before \texttt{\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 \texttt{manyfoot} and \texttt{bigfoot}) providing footnote features beyond standard \LaTeX{}.

2. \texttt{\lipsum[\langle opt-arg \rangle]} in main text produces a different number of paragraphs . . .

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

4. Much of the code is “guessed” without complete knowledge of \TeX{} internals and without having tested many possible cases.

5. \textit{Local} switching to “pagewise” numbering won’t be possible for a while; we rather assume that you \textit{always} want “pagewise” numbering.

6. Nothing has been tried to offer choices about the \textit{style} of numbering footnotes.
2 Implementation

2.1 Terms

“OTR” is short for “output routine”, “MVL” is short for “main vertical list”.

2.2 Basic Strategy

\LaTeX{}’s \texttt{@footnotetext} writes the footnote text into the insertion register. For numbering the footnote lines, we here do not execute this \texttt{@footnotetext} immediately after placing \texttt{@footnotemark} but postpone its \texttt{\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 \texttt{\insert} register.

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

2.3 Package Options

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

\begin{verbatim}
\newif\if@FNLN@check@
\DeclareOption{check-latex}{\@FNLN@check@true}
\ProcessOptions
\end{verbatim}

2.4 Footnote Commands

2.4.1 Standard Footnotes

The following macro \texttt{\FNLN@ltx@fntext} is a copy of \LaTeX{}’s \texttt{@footnotetext} that we are varying. It may be used for a check if the \texttt{@footnotetext} that fnlineno.sty encounters is the one expected (\texttt{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 processing footnotes.

\begin{verbatim}
\if@FNLN@check@
  \long\def\FNLN@ltx@fntext#1{\insert\footins{\reset@font\footnotesize
  \interlinepenalty\interfootnotelinepenalty
  \splitopkip\footnotesep
  \splitmaxdepth \dp\strutbox \floatingpenalty \@MM
  \hspace{\linewidth} \@parboxrestore
  \protected@edef\@currentlabel{\csname p@footnote\endcsname\@thefnmark}
  \insert\footins{#1}}}
  \else
  \FNLN@ltx@fntext{#1}
\fi
\end{verbatim}
2.4.2 Modifying Footnote Commands

In order to number \footnote lines and make \linelabel available in footnotes, 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...

\let\FNLN@text@footnotetext
\def\@footnotetext{% 
  \ifLineNumbers \expandafter \FNLN@text 
  \else \expandafter \FNLN@@text 
  \fi}
\def \FNLN@text {
  \vadjust{\penalty-\FNLN@M@swap@codepen}%

Standard \LaTeX’s \footnotetext expands \thefnmark to produce the footnote 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 (\protected) 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⟩\elt⟨text⟩\elt’. \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 \protectededef:

\let\@protect\protect
\let\protect\unexpandable@protect
\afterassignment\restore@protect
\long \edef \@tempa #1{%
\noexpand\g@addto@macro \noexpand\FNLN@list {%
\noexpand\thefnmark \noexpand\elt #1 \noexpand \elt} %
}%

... issuing ‘\g@addto@macro \FNLN@list {{⟨mark⟩}\elt⟨text⟩}\elt’...
\@tempa % reads arg
2 IMPLEMENTATION

Here we initialize \lst{FNLM@list}:

\let\FNLM@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).

\if@FNLM@check@
  \def\FNLM@lno@output {%
    \LineNoTest
    \if@tempswa
      \ifnum\outputpenalty=-\@Mllbcodepen
        \WriteLineNo
      \else
        \ifnum\outputpenalty=-\@Mppvacodepen
          \PassVadjustList
        \else
          \LineNoLaTeXOutput
        \fi
      \fi
    \else
      \MakeLineNo
    \fi
  \fi
  \fi

The “signal penalties” used here are

\mathchardef\FNLM@M@llbl@codepen=11111
\mathchardef\FNLM@M@ppva@codepen=11112

Their names should mean “\linelabel code penalty” and “\PostponeVadjust code penalty.”

\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.

\let\TheLineNoLaTeXOutput\LineNoLaTeXOutput\LineNoLaTeXOutput

2.5.2 Tools for Temporary Parameter Changes

\GStoreReg{⟨register⟩} (or \GStoreReg{⟨register⟩})
when ⟨register⟩ is a single token—‘\count0’ being a counterexample…—stores the current content of ⟨register⟩ (globally) as an internal macro so that it can
be restored later by
\RestoreReg\langle\text{register}\rangle \quad \text{(or \RestoreReg\langle\text{register}\rangle)}

or globally by
\GRestoreReg\langle\text{register}\rangle \quad \text{(\GRestoreReg\langle\text{register}\rangle)}

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

\GStoreSetReg\langle\text{register}\rangle\langle\text{value}\rangle assigns \langle\text{value}\rangle to \langle\text{register}\rangle (locally) after executing \GStoreSet. \GStoreGSetReg does the same \textit{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\text{register}\rangles …)

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 \texttt{pagebreak} in footnote texts, using a flag \texttt{\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 \texttt{\% v0.41}

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

83 \newsavebox\FNLN@holdft \texttt{\% v0.41}

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

84 \renewcommand*{\LineNoLaTeXOutput}{%  
85 \ifnum\outputpenalty=\FNLN@M@swap@codepen  
86 \SwapFootnoteMain  
87 \else
\ifnum\outputpenalty=-\FNLN@M@insert@codepen
\InsertFootnote
\else
\if\FNLN@sw@ \% v0.41
% \showthe\outputpenalty \% 2010/12/20
\global\setbox \FNLN@holdft \vbox{%
\unvbox\FNLN@holdft
\InsertFootnote
\else
% \showthe\outputpenalty \% 2010/12/20
\global\setbox \FNLN@holdft \vbox{%
\unvbox\FNLN@holdft
\InsertFootnote
\fi
\fi
\fi
\fi}

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!

\unvbox\@cclv

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

\penalty\outputpenalty}%
% TODO reset page book-keeping!? \% v0.41
\else
\TheLineNoLaTeXOutput \% "the real \LineNoLaTeXOutput"
\fi
\fi
\fi
\}

An idea: Instead of so many \ifnum, use

\csname chars\endcsname the\outputpenalty

... 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 \FNLN@holdcol.

\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:

\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:

\GStoreGSetReg\topskip\z@skip
\(\text{\noindent lineskip as well as setting \topskip locally instead fails \ldots according to \showlists\ldots\)} \Reset\pagegoal (why doesn’t it switch to \vsize = \maxdimen automatically?), \pagetotal and the other “special dimens” (\TeXbook p. 271; rather experimental \ldots I think it is important to restore them later \ldots)

\GStoreSetReg\pagegoal \vsize
\GStoreSetReg\pagetotal\z@\z@
\GStoreSetReg\pagestretch\z@
\GStoreSetReg\pagefillstretch\z@
\GStoreSetReg\pagefillstretch\z@
\GStoreSetReg\pageshrink\z@
\GStoreSetReg\pagedepth\z@

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

\reset@font\footnotesize
\interlinepenalty\interfootnotelinepenalty
\noindent L\TeX’s split things here are relevant at \insert\footins only: (\text{TODO}!?)

\% \sPLIT\topskip\footnotesep
\% \sPLIT\vsize \dp\strutbox \floatingpenalty \z@\z@\z@\z@
\hspace\columnwidth \@parboxrestore

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

\expandafter \FNLN@typeset \FNLN@list \z@
\% \showthe\vsize

... so a \vsize assignment without \global is noted here, and an analogous \topskip assignment is not!? \text{TODO} \ldots
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:

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

This was our own, and next \TeX{} continues:

\protected@edef \@currentlabel{% 
  \csname p@footnote\endcsname \@thefnmark
}%
\color@begingroup

We insert starting the lineno settings ... 

\lnumbers \setfootnotelinenumbers \% 2010/12/25

... \TeX{} again (v0.41 exports dealing with closing \par to finstrut.sty):

\@makefntext{% 
  \rule\z@ \footnotepsep \ignorespaces
    #2\par 
  \@finalstrut \strutbox
}%
\color@endgroup

Now we trigger the “swap back slot” of the OTR:

\penalty-\FNLN@M@insert@codepen
\}
\RequirePackage{finstrut}

2.5.5 \texttt{\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 \TeX{}’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:

\newcommand*{\InsertFootnote}{% 
  \GRestoreReg\topskip \ GRestoreReg\vsize
With v0.5, global settings for “pagewise” numbering must be restored:

\unsetfootnotelinenumbers
}

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” desirerment 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 \((n, \ n + 1, \ m, \ or \ m + 1 \ldots)\).

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. \texttt{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 \texttt{\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 \( \{n, \ldots, k\} \). 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 \(\{N, \ldots, K\}\) 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 \texttt{.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 \texttt{.aux} file that the run creates, a two-parameter macro \texttt{\@LN}. With \texttt{fnlineno.sty}, there will be new \texttt{\@FLN} entries of the same type. These \texttt{.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 \texttt{\ref} and \texttt{\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 \linelabels 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.

```
\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.
\def \FNLN@first@numbered \LN@Pfirst

(oh, it must be \def here to recognize the change . . . ). This must be changed by \setfootnotelinenumbers (\let them, 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):

\def \FNLN@last@numbered \LastNumberedPage

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

\def \FNLN@last@numbered \LastNumberedPage

\global \let \FNLN@last@numbered \LastNumberedPage

\FNLN{\{names\}\{last-numbered\}\{line\}\{page\}}

generalizes lineno.sty’s \LN{\{line\}\{page\}} to re-implement it. There is an additional parameter argument \textit{\textbackslash names} for choosing name spaces and a parameter \textit{\textbackslash last-numbered} for choosing the macro storing the “last numbered page.” (An argument without braces expects a macro name.)

\newcommand* \FNLN [4]{{%\expandafter \@@LN \csname #1#4C\LN@column \expandafter \endcsname \csname #1O#4\endcsname \{#3\}\{#4\}\{#1\}\{#2\}}

As in lineno.sty \LN calls \FLN, a new variant of \FLN is called by \FNLN here, but it gets one additional parameter for passing \textit{\textbackslash names} and another for passing \textit{\textbackslash last-numbered} from \FLN. So the new syntax is

\FNLN{\{info\}\{first-page-line\}\{line\}\{page\}\{names\}\{last-numbered\}}

\renewcommand* \FLN [6]{{%
\expandafter \@@@LN \csname \textbackslash info\textbackslash first-page-line\textbackslash page\textbackslash names\textbackslash last-numbered\endcsname}}

\newcommand* \FLN [6]{{%
\expandafter \@@@LN \csname \textbackslash info\textbackslash first-page-line\textbackslash page\textbackslash names\textbackslash last-numbered\endcsname}}

\renewcommand* \FLN [6]{{%
\expandafter \@@@LN \csname \textbackslash info\textbackslash first-page-line\textbackslash page\textbackslash names\textbackslash last-numbered\endcsname}}

As in lineno.sty \LN calls \FLN, a new variant of \FLN is called by \FNLN here, but it gets one additional parameter for passing \textit{\textbackslash names} and another for passing \textit{\textbackslash last-numbered} from \FLN. So the new syntax is

\FNLN{\{info\}\{first-page-line\}\{line\}\{page\}\{names\}\{last-numbered\}}

\renewcommand* \FLN [6]{{%
\expandafter \@@@LN \csname \textbackslash info\textbackslash first-page-line\textbackslash page\textbackslash names\textbackslash last-numbered\endcsname}}

\newcommand* \FLN [6]{{%
\expandafter \@@@LN \csname \textbackslash info\textbackslash first-page-line\textbackslash page\textbackslash names\textbackslash last-numbered\endcsname}}

As in lineno.sty \LN calls \FLN, a new variant of \FLN is called by \FNLN here, but it gets one additional parameter for passing \textit{\textbackslash names} and another for passing \textit{\textbackslash last-numbered} from \FLN. So the new syntax is

\FNLN{\{info\}\{first-page-line\}\{line\}\{page\}\{names\}\{last-numbered\}}

\renewcommand* \FLN [6]{{%
\expandafter \@@@LN \csname \textbackslash info\textbackslash first-page-line\textbackslash page\textbackslash names\textbackslash last-numbered\endcsname}}

\newcommand* \FLN [6]{{%
\expandafter \@@@LN \csname \textbackslash info\textbackslash first-page-line\textbackslash page\textbackslash names\textbackslash last-numbered\endcsname}}
2 IMPLEMENTATION

\\def#1\{lastLN{#3}\}firstLN{#3}]{% 
  \pageLN{#4}\{\OLN@column\}{#2}\nextLN{relax}%
\} 
\else\def\lastLN##1\{\noexpand\lastLN{#3}}\% 
\def#1\{#1}\% 
\fi\def#6\{#4C\OLN@column\}}
\@onlypreamble\@@LN
\alineno.sty's \@@@LN does not need any adjustment.
\alineno.sty's \@LN{⟨line⟩}{⟨page⟩} is reimplemented as
\def \OLN \{\@FNLN{LN@P}\LastNumberedPage\}
—so \OLN really does the same as before, including name spaces.
\@FNLN{⟨line⟩}{⟨page⟩} is the other special case of the new \@FNLN—an F precedes the earlier names, and \@FNLN@last@numbered is the storing macro initialized above:
\def \@FNLN \{\@FNLN{FLN@P}\@FNLN@last@numbered\}
For logging, we make both unexpandable:
\% \AtBeginDocument{\let\OLN\relax \let\@FNLN\relax}
\ldots but this way nothing appears in the file!? TODO \ldots
\@onlypreamble\OLN \@onlypreamble\@FNLN
For reading the .aux finally, we do what \alineno does with \OLN:
\AtEndDocument{\let\@FNLN\gobbletwo}

2.6.5 Tool for Reusing Global Operations with Macros
\alineno.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 ⟨ln-macro⟩ for ⟨fln-macro⟩, we \global\let⟨ln-macro⟩⟨fln-macro⟩, apply the operations, and finally \global\let⟨fln-macro⟩⟨ln-macro⟩. However, we are not only interested in how ⟨fln-macro⟩ is changed this way, rather ⟨ln-macro⟩ also is used as input for some operations, and we can choose which ⟨fln-macro⟩ should be used as input. To switch from working on/with ⟨fln-1⟩ to ⟨fln-2⟩ using ⟨ln-macro⟩ with an option to use ⟨fln-1⟩ later again, a tool \GStoreUse[⟨ln-macro⟩]⟨fln-1⟩⟨fln-2⟩ is provided (should render later switchings much better readable):
\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 \texttt{pagewise} must be global (I can’t find something useful using \texttt{aftergroup} quickly). Therefore, I render \texttt{lineno}'s \texttt{\setpagewisenumbers} acting globally:

\begin{verbatim}
\renewcommand*{\setpagewisenumbers}{%
\global\let \theLineNumber \thePagewiseLineNumber
\global\let \c@linenumber \c@pagewiselinenumber
\global\let \makeLineNumber \makePagewiseLineNumber
%
}

I just force this, hehe . . .
\end{verbatim}

\texttt{\setpagewiselinenumbers}

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

\begin{verbatim}
\newcount\c@footnotelinenumber
\FNLN@@cache stores \texttt{\NumberedPageCache} as from “main” mode:
\end{verbatim}

\texttt{\FNLN@cache} stores \texttt{\NumberedPageCache} as from “footnote” mode; its initial content is the counterpart or analogue to \texttt{\LN@Pfirst}:

\begin{verbatim}
\def \FNLN@cache {\FLN@Pfirst}
\end{verbatim}

\texttt{\FNLN@foot@cache} and \texttt{\FNLN@main@cache} switch \texttt{\NumberedPageCache}:

\begin{verbatim}
\def \FNLN@foot@cache {\GStoreUse \NumberedPageCache \FNLN@@cache \FNLN@cache}
\def \FNLN@main@cache {\GStoreUse \NumberedPageCache \FNLN@cache \FNLN@@cache}
\end{verbatim}

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

\begin{verbatim}
\global\let \FNLN@labels \@empty
\end{verbatim}

\texttt{\FNLN@vadjusts} will be the counterpart to \texttt{\lineno.sty}'s \texttt{\@LN@vadjustlist}:

\begin{verbatim}
\global\let \FNLN@vadjusts \@empty
\end{verbatim}

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

\begin{verbatim}
\newcommand* \setgetfootnotelinenumbers {%
Change of \texttt{\theLineNumber} is omitted as we are \emph{reading}, not writing a label.

\begin{verbatim}
\let\c@linenumber\c@footnotelinenumber
\let\makeLineNumber\makeFootnoteLineNumber
\end{verbatim}

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

\begin{verbatim}
\let\FNLN@first@numbered\FLN@Pfirst
\let\FNLN@finish@numbered\FLN@Pfirst
\end{verbatim}

\texttt{\setfootnotelinenumbers} performs all the settings for typesetting footnotes in line numbering mode \emph{globally}, including storing results from typesetting main text:

\begin{verbatim}
\newcommand* {\setfootnotelinenumbers}{%\globaldefs\@ne
\end{verbatim}

The previous line also renders \texttt{\setgetfootnotelinenumbers} global:

\begin{verbatim}
\setfootnotelinenumbers
\end{verbatim}

\texttt{\theLineNumber} is used for \texttt{\linelabel} entries. \texttt{\thePagewiseLineNumber} is replaced by \texttt{\theFootnoteLineNumber}:

\begin{verbatim}
\let\theLineNumber\theFootnoteLineNumber
\end{verbatim}

Logging to \texttt{.aux}:

\begin{verbatim}
\def\FNLN@log{\string\@FLN}%
\end{verbatim}

Starting range search: \texttt{\NumberedPageCache}

\begin{verbatim}
\FNLN@foot@cache
\end{verbatim}

Reusing \texttt{lineno}'s task list operations:

\begin{verbatim}
\GStoreUse\@LN@labellist\FNLN@labels\FNLN@labels
\GStoreUse\@LN@vadjustlist\FNLN@vadjusts\FNLN@vadjusts
\globaldefs\z@
\end{verbatim}

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 \texttt{\setgetpagewiselinenumbers}:

\begin{verbatim}
\newcommand* {\setgetpagewiselinenumbers}{%\globaldefs\@ne
\end{verbatim}

\begin{verbatim}
\let\FNLN@first@numbered\LN@Pfirst
\let\FNLN@finish\LN@Pfirst
\end{verbatim}

\begin{verbatim}
\end{verbatim}
stores the “current” page with footnote lines and loads the “most recent” page with main text lines—and more . . .:

\newcommand*{\unsetfootnotelinenumbers}{% \gdef{\FNLN@log}{\string@LN}% \FNLN@main@cache}

Task lists:
\GStoreUse{\@LN@labellist}{\FNLN@labels}{\FNLN@@labels}
\GStoreUse{\@LN@vadjustlist}{\FNLN@vadjusts}{\FNLN@@vadjusts}
\globaldefs{\@ne}{\setgetpagewiselinenumbers}{\globaldefs{\z@}{%% v0.53}
\setpagewiselinenumbers
}

\makeFootnoteLineNumber actually only copies \makePagewiseLineNumber, different results are obtained by 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).

\@ifdefinable{makeFootnoteLineNumber}{\let{makeFootnoteLineNumber}{makePagewiseLineNumber}}

2.6.7 Logging
\logtheLineNumber is redefined to log both main text and footnote line numbers.
\def{\logtheLineNumber}{% \protected@write{\@auxout}{% \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:
\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):
\def{\FNLN@cache}{\FLN@Pfirst}
\texttt{\renewcommand* \testFirstNumberedPage [1] {\%}} \texttt{\ifnum#1 > \c@linenumber \%}
\texttt{\let\nextLN \@gobble \%}
\texttt{\def\pageLN {\gotNumberedPage {#1}} \%}
\texttt{\fi} \%

\texttt{\renewcommand* \gotNumberedPage {\%}} \texttt{\oddNumberedPage false \%}
\texttt{\ifodd \if@twocolumn #3 \else \relax \oddNumberedPage true \fi}
\texttt{\advance \c@linenumber \@ne \%}
\texttt{\ifcolumnwiselinenumbers}
\texttt{\let \firstLN \subtractlinenumberoffset {#1} \%}
\texttt{\else}
\texttt{\let \firstLN \@gobble \%}
\texttt{\def \pageLN {\subtractlinenumberoffset {#1} {#2} {#3} {#4}} \%}
\texttt{\fi \%}
\texttt{\show \FNLN@finish \%}
\texttt{\FNLN@finish {\%}} \texttt{\gobbletwo \%}

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

\texttt{\newcommand* \FNLN@add [2] {\%}} \texttt{\expandafter \let \expandafter \@tempa \csname LN@P#1 C#2 \endcsname \%}
\texttt{\ifix \@tempa \relax \%}
\texttt{\else}
\texttt{\advance \c@linenumber \@ne \%}
\texttt{\ifcolumnwiselinenumbers}
\texttt{\let \firstLN \subtractlinenumberoffset \%}
\texttt{\else}
\texttt{\let \firstLN \@gobble \%}
\texttt{\def \pageLN {\subtractlinenumberoffset {#1} {#2} {#3} {#4}} \%}

... rather assuming \texttt{\realpagewiselinenumbers}.
2 IMPLEMENTATION

2.6.9 Referencing

Now that we are using two separate counters for main text lines and footnote lines (v0.5), correct references to footnote lines using \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.

\theWiseLineNumber{(trans)} leaves a \protect call to a one-parameter macro \trans in the .aux file:

\newcommand* \theWiseLineNumber [1]{\protect #1{\the\c@linenumber}}
\getwiselinenumber{(choice)}{(integer)} executes \choice before applying \testNumberedPage to \integer—within a local group:

\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.

\ifx#1\relax % to main text
\if@FNLN@sw@ % from footnote
\setgetpagewiselinenumbers
\FNLN@main@cache
\let \FNLN@restore@cache \FNLN@foot@cache
\fi
\else % to footnote
\if@FNLN@sw@ \else % from elsewhere
\fi
#1%
\texttt{\textbackslash getpagewiselinenumber} doesn’t need any \texttt{choice}—we assume that the label was written in the default \texttt{pagewise} mode (but it is difficult, though, \texttt{\relax} is essential!):

\begin{verbatim}
% \renewcommand* \getpagewiselinenumber \{\getwiselinenumber\relax\} %!!
\AtBeginDocument{\
def \getpagewiselinenumber \{\getwiselinenumber\relax\} \% sic!\let \@EN\getpagewiselno \getpagewiselinenumber}
\end{verbatim}

For \texttt{\thePagewiseLineNumber} \texttt{\langle trans \rangle} is \texttt{\getpagewiselinenumber}:

\begin{verbatim}
\renewcommand* \thePagewiseLineNumber \{\% \theWiseLineNumber\getpagewiselinenumber\}
\end{verbatim}

\texttt{\getfootnotelinenumber\langle integer\rangle} considers \texttt{\langle integer\rangle} the absolute number of a \texttt{footnote} line. The \texttt{\langle choice\rangle} therefore is \texttt{\setgetfootnotelinenumbers}:

\begin{verbatim}
\newcommand* \getfootnotelinenumber \{\% \getwiselinenumber\setgetfootnotelinenumbers\}
\end{verbatim}

Finally, \texttt{\theFootnoteLineNumber} is how \texttt{\linelabel} refers to a \texttt{footnote} line. \texttt{\theWiseLineNumber} is called with \texttt{\langle trans \rangle} being \texttt{\getfootnotelinenumber}:

\begin{verbatim}
\newcommand* \theFootnoteLineNumber \{\% \theWiseLineNumber\getfootnotelinenumber\}
\end{verbatim}

\section{Leaving the Package File}

\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 \TeX\book. 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

v0.1 2010/12/08 very first, \linelabel works in footnote
SENT TO Christian, problems with "long" footnotes

v0.2 2010/12/08 corr. "manifoot"
2010/12/09 moving doc. from .tex to here,
different doc. sectioning;
\footnotetext modified (user feature!);
\docclearpage NOT modified!; \if@FNLN@placing@
2010/12/10 ignore dummy footnote split;
\FNLNpar, \AutoPars, \ExplicitPars,
more on limitations
2010/12/11 more trying, almost anew ...
JUST STORED

v0.3 2010/12/12 new approach, removed much before proceeding
2010/12/13 -- this was putting \box\footins onto MVL,
bad with those penalties
JUST STORED

v0.4 2010/12/14 another new approach:
typeset footnote on MVL immediately --
described strategy
2010/12/15 ... continued, choice of hooking into \output
(...swap...)  
2010/12/16 ... continued; rearranged sections ... 
\FNLN@fntext vs. ...ltex... 
2010/12/17 success with \pagegoal ...; \GStoreReg etc.; ...\fntext shortened 
2010/12/18 another two limitations: \pagebreak in fn., 
guessed/tested; another note to <register>; 
ack. Christian; directed -> organized!
SENT TO Christian/Stephan

v0.41 2010/12/19 support of \pagebreak with \if@FNLN@sw@ etc.;
TODO on lists of <register>s

2010/12/20 debugging: \if...true; \setbox...ft;
\@finalstrut in vmode exported to finstrut.sty;
notes on how v0.41 still fails with \pagebreak
2010/12/21 additional notes on *two* \pagebreak's

v0.5 2010/12/21 restructuring doc., check@latex@ -> check@,
own account of lineno's pagewise mode
2010/12/22 ... continued ...
2010/12/23 ... continued ...
2010/12/24 ... continued ...
2010/12/25 moved this to pulineno, replaced ...
more on \FNLN@typeset, + \setfootnotelinenumbers
2010/12/26 new summary of implementation,
rearranged code sections; logging settled
v0.51 2010/12/27 "build" settled, typesetting, logging reformed;
ack.s: "recall"; all settings global,
"public" works
JUST STORED, MARGINAL NUMBERS OK,
\linelabel in footnote broken
[2010/12/28]
v0.52 2010/12/28 own label and vadjust lists for footnotes;
local settings for referencing,
tool and care for global changes (...Cache)
(TODD write ready in .aux? needs another run)
\linelabel's ok, MARGINAL NOTES MAIN BROKEN
v0.53 2010/12/28 debugging; OK; minor doc. modifications;
less "limitations"; \[\smallskipamount
TO CHRISTIAN 2010-12-29
v0.54 2010/12/31 typo options; \FNLN@text without arg,
\getpagewiselinenumber with ednotes
2011/01/01 \FNLN@cache, \FNLN@cache initialized;
doc. "Typesetting Stage" qualification
2011/01/02 that qualification was wrong
2011/01/03 samepage@hook
TO CHRISTIAN SAME DAY
v0.55 2011/01/04 samepage@hook emptied here as well;
2011/01/06 edited version history
2011/01/07 note on \if@FNLN@sw@ with v0.5;
finally without support for samepage@hook!
note on \testLastNumberedPage
PART OF EDFN RELEASE r0.5 (together with edfnotes v0.2)
v0.55a 2011/02/09 corr. owner; "Limitations" updated; \pagebreak