edfnotes.sty

—

Critical Annotations to Footnote Lines*

Uwe Lück†

February 22, 2011

Abstract

edfnotes.sty extends ednotes.sty[1] so that you can refer even to *footnotes* of the edited work by line numbers, building on the accompanying fnlineno package in the lineno bundle.[2] ednotes.sty has addressed "scholarly" critical editions of (hand-written) manuscripts. edfnotes additionally supports critical editions of (printed) *works with footnotes.* The package was developed for an edition of Bernard Bolzano's *Paradoxien des Unendlichen* by Prof. Dr. Dr. Christian Tapp—a work with some very long footnotes.

As to *implementation*, edfnotes modifies ednotes' annotation commands and label-test mechanism so that the latter does not break and the former place critical notes on the pages and in the order as one naturally expects (not so easy with footnotes). Certain core parts of ednotes have been re-implemented entirely, this may later migrate into ednotes itself. It had been hoped that edfnotes could build on the bigfoot package.[3] This might have improved (automatic) page breaking and placement of original footnotes and critical annotations (whereas at present, some page breaks need manual trial-and-error solutions). However, analysis of lineno and ednotes for extending them, as well as many unexpected difficulties, exhausted project resources too early. David Kastrup's `\MakeSorted` (or `\MakeSortedPerPage`, actually from perpage)[4] has here been implemented by `\insert`ing annotations to footnotes at the last main text line of a page only, hooking into lineno's numbering mechanism—a nice ("leightweight"—?) alternative to David Kastrup's approach.

Keywords: critical editions; footnotes

*This document describes version v0.6b of edfnotes.sty as of 2011/02/16.

†http://contact-ednotes.sty.de.vu

[1] http://ctan.org/pkg/ednotes

[2] http://ctan.org/pkg/lineno

[3] http://ctan.org/pkg/bigfoot

[4] http://ctan.org/pkg/perpage

Contents

1 Usage and Features

1.1 Package File Header (Legalize)

```
1  \NeedsTeXFormat{LaTeX2e}[1994/12/01]
2  \ProvidesPackage{edfnotes}[2011/02/16 v0.6b
3                             annotations to footnote lines (UL)]
4
5  %% Copyright (C) 2010 2011 Uwe Lueck,
6  %% http://www.contact-ednotes.sty.de.vu
7  %% -- author-maintained in the sense of LPPL below --
8  %%
9  %% This file can be redistributed and/or modified under
10 %% the terms of the LaTeX Project Public License; either
11 %% version 1.3c of the License, or any later version.
12 %% The latest version of this license is in
13 %%     http://www.latex-project.org/lppl.txt
14 %% We did our best to help you, but there is NO WARRANTY.
```

```
15  %%
16  %% Please report bugs, problems, and suggestions via
17  %%
18  %%   http://www.contact-ednotes.sty.de.vu
19  %%
```

This work has been supported by the Deutsche Forschungsgemeinschaft (DFG), organized by Prof. Dr. Dr. Christian Tapp at Ruhr-Universität Bochum, Germany. Christian also participated immensely in development by creating really small failure examples for debugging. Completing the work was possible thanks to a loan from my mother, Mrs. Renate Lück.

1.2 Known Issues

Please note the following difficulties you may experience with edfnotes, and what to do about them:

1. In certain situations, "**hanging**" of the `latex` run has been observed. As to *implementation*, it has not been analysed completely yet (TODO), but it occurred with `\marginpar` and printing the second part of a footnote before its first part—a well-known LaTeX bug, "**footnote placement disaster**." However, this situation usually does not lead into "hanging," rather replacing `=` by `>` in the definition of `\EFN@annot` with package version v0.6 (to keep "displaced" annotations from wandering to the last page) seems to cause the hang when a "footnote placement disaster" occurs (where the footnote has annotations).

   You must be careful with marginals anyway—with critical editions. There also is the problem with placement in the wrong margin. Mistakes of the named kinds occur rather *usually* in critical editions. With *annotations to footnotes* things may become even worse, since placement restrictions ("start note on same page as target") may become difficult or impossible to satisfy.[5]

   Often, just manual `\pagebreak`s (in the last line of a page—when near completion of the work) help.

   The mparhack package[6] at least ensures that marginals appear in the appropriate margin.

   "Footnote placement disasters" can be avoided (at some cost) by the tamefloats package[7] (helped sometimes, did not help with *Paradoxien des Unendlichen*). Another possibility for avoiding footnote placement disasters due to `\marginpar` is using `\marginnote` from the marginnote package[8] instead.

[5] These advices should appear in ednotes already—TODO!
[6] `http://ctan.org/pkg/mparhack`
[7] `http://ctan.org/pkg/tamefloats`
[8] `http://ctan.org/pkg/marginnote`

However, \marginnote behaves badly when, e.g., page breaks in two sources appear in the same line of the edition. On the other hand, \marginpar "moves" the second marginal, which is not much better. I would prefer a "manual" solution: inserting the page break symbols within the line manually and produce a single \marginpar or \marginnote entry for both of them. (Actually, this could be automated within lineno.)

2. **Page break oscillations:** ednotes.sty provides a message about page break oscillations that gives advice how to fix them. With annotations to footnotes, that advice may help only little. Besides fixing the page break in main text, it may be needed to place \pagebreak (or \warningpagebreak) commands in footnotes too. While the oscillation of main text lines is fixed (for the page about the message reports), page breaks of footnotes may keep oscillating. I was successful in a test case (that giant footnote in *Paradoxien des Unendlichen*) by placing a \pagebreak in the footnote text quite early first and then moved it down line by line, until the result became bad. At that point, I chose the previous position of \pagebreak. The result is presented as PdUsample.pdf.

3. For **other Limitations**, please see the section in fnlineno.pdf of this title.

1.3 Installing and Calling

The file edfnotes.sty is provided ready, installation only requires putting it somewhere where TEX finds it (which may need updating the filename data base).[9]

As usually, edfnotes.sty is loaded by \usepackage{edfnotes} below the \documentclass line (after loading ednotes) and before \begin{document}. edfnotes does not offer any package options.

This should suffice just to get the advertised functionality. There are *two* user commands \ForceFootnoteAnnotations and \clearfootnoteannotations described in section 2.5.4.

Minimum functionality of edfnotes requires that ednotes.sty, fnlineno.sty, and lineno.sty are installed and "visible." For more functionality, see the descriptions of both bundles.[10][11].

[9] http://www.tex.ac.uk/cgi-bin/texfaq2html?label=inst-wlcf
[10] http://ctan.org/pkg/ednotes
[11] http://ctan.org/pkg/lineno

2 Implementation

2.1 Loading Required Package

The package builds on fnlineno's ability to number footnote lines and processing `\linelabel` there. So we load it. We need a version without earlier tries at supporting edfnotes.sty v0.2:

```
20  \RequirePackage{fnlineno}[2011/01/07]
```

2.2 Overview of Processing Levels

ednotes' and edfnotes' apparatus entries are handled by the manyfoot package. An entry ⟨*insert*⟩ for apparatus level ⟨*note-fam*⟩ is passed to manyfoot by manyfoot's command `\Footnotetext`⟨*note-fam*⟩`\@empty{`⟨*insert*⟩`}`.

In terms of TEX's hard-wired mechanisms, such a `\Foonotetext` command issues an `\insert` command. `\insert` is a so-called "primitive," hard-wired command. It typesets the entry ⟨*insert*⟩ and stores the result internally, and in choosing the next page break, TEX will try to start the typesetting result of ⟨*insert*⟩ on the "current" page.

TEX is not so well-prepared (**hard-wired**) for placing a footnote ⟨*ff*⟩ (its beginning) intended to refer to a passage in another footnote ⟨*f*⟩ on the same page as that passage (starts). Our **hard-weird** macros build an own mechanism using the `.aux` file in order to manage this. A footnote ⟨*ff*⟩ to a footnote ⟨*f*⟩ is not directly passed to manyfoot, such "secondary" footnotes ⟨*ff*⟩ are rather withhold and collected in a list ⟨*list*⟩ from which edfnotes will later decide what to do with them. The goal is to find a good point for finally passing ⟨*ff*⟩ to manyfoot such that TEX will be able to place ⟨*f*⟩ and ⟨*ff*⟩ appropriately.

In the most basic application of edfnotes, there is *one* critical apparatus referring both to the main text and to the footnotes of an edited text. We want that, in the apparatus, annotations to *main text* lines precede annotations to *footnote* lines. To achieve this, the idea is to issue the `\Footnotetext` commands for annotations to footnotes only at *last main text lines* of pages.

Unfortunately, sometimes the `.aux` file provides either *no* or *wrong* information for identifying that last line. Therefore, we sometimes *skip* the item-by-item procedure that tries to pair ⟨*ff*⟩ with ⟨*f*⟩, and instead somewhere pass *all* the ⟨*ff*⟩ items in that list ⟨*list*⟩ to manyfoot and TEX's primitive `\insert` command. This will be just a kind of "emergency" routine, considered necessary temporarily, while hoping that the required and correct information will appear in the `.aux` file some runs later.

2.3 Postponing Annotations

2.3.1 Goal and Strategy

Version v0.2 of edfnotes used `\@EN@hookfn{`⟨*insert*⟩`}`—that is called from ednotes' `\@EN@putdown`—in order to intercept manyfoot's `\FootnotetextA` etc. commands that execute TEX's `\insert` commands for ednotes' annotations. `\@EN@hookfn` has the trivial meaning `\@firstofone` in ednotes and is only activated by edtable.sty (lineno bundle) in order to collect annotations to a tabular environment at the latter's "footnote hook" that finally executes the `\insert` commands. These annotations are inserted to the *current* page, i.e., all the annotations collected there start on the same page.

This behaviour (starting on the same page) is in general not adequate with annotations to footnotes when the latter are split and continued at later pages. If a footnote starts at page p, is continued on page $p+1$, and a passage of it on page $p+1$ gets a critical annotation, that annotation should start on page $p+1$, not on page p. The footnote may also get an annotation to its page p part, that one should start at page p indeed. The mechanism we have to introduce here must be able to tell these two cases apart. For each annotation to a footnote, it must retrieve the page which the annotation refers to.

Interception at ednotes' `\@EN@hookfn` now becomes too clumsy. We redefine

`\@EN@putdown{`⟨*id*⟩`}{`⟨*note-fam*⟩`}{`⟨*lemma-tag*⟩`}{`⟨*note*⟩`}`

entirely, replacing its former `\@EN@hookfn` section by a new macro

`\@EN@appentry{`⟨*id*⟩`}{`⟨*note-fam*⟩`}{`⟨*lemma-tag*⟩`}{`⟨*note*⟩`}`

i.e., taking the same arguments as `\@EN@putdown`. ⟨*id*⟩ is a string that is used to write commands `\newlabel{-`⟨*id*⟩`}{`⟨*l/p-b*⟩`}`, `\newlabel{+`⟨*id*⟩`}{`⟨*l/p-e*⟩`}`, and `\newlabel{`⟨*id*⟩`}{`⟨*note-page*⟩`}` to the `.aux` file. From these commands, the next run builds a macro containing placement informations for that annotation with ID ⟨*id*⟩. ⟨*l/p-b*⟩ contains the page on which the annotated passage starts. This is where the annotation *should* start as well. It just contains the "public" ("relative") page number, as opposed to "absolute" page numbers that lineno.sty traces through `\c@LN@truepage`. Well, the minute chance of failure arising from this difference should be acceptable for the present attempt at equipping edfnotes with functionality adequate for real life.

2.3.2 Re-implementation of `\@EN@putdown`

`\@EN@putdown` ...

```
\long\def\@EN@putdown#1#2#3#4{%
% #1 label, #2 note family, #3 lemma tag, #4 note.
% \long note/lemma!?
  \linelabel{-#1}% Ensures hmode.
```

`\insert` apparatus entry:

```
  \@EN@appentry{#1}{#2}{#3}{#4}%
```

Lemma in main text:

```
    \nobreak \hskip\z@skip %% 2006/01/12 cf. german.sty: \allowhyphens
    \@EN@lemmaarg %% Should not end with space so \linelabel...
    %% \allowhyphens not needed here (tested) 2006/01/12
    \linelabel{+#1}%
    % \csname #1rightmark\endcsname %% Suggested.
  }
```

`\@EN@appentry` ...

```
  \def \@EN@appentry #1#2#3#4{%
    \csname Footnotetext#2\endcsname\@empty{%
      \csname#2notefmt\endcsname
      \let\@currentlabel\@empty
      \label{#1}% Need page number of note.
      \let\nopunct\@gobble % or by \if...
```

Print line numbers:

```
      \@ifundefined{r@#1}%
        {\differentlines\@EN@unknown\@EN@unknown}%
        {\let\@EN@incomplete@ref\relax
         \@EN@xpxpxp \@EN@extract\csname r@#1\endcsname
           \@EN@incomplete@ref\@EN@incomplete@ref
             \@EN@incomplete@ref\@EN@incomplete@ref
         \ifx\@EN@elemp\@EN@Incomplete
           \G@refundefinedtrue
           \let\@EN@incomplete@ref\@EN@unknown
         \fi
         \@tempswatrue
         \ifx\@EN@bleml\@EN@eleml \else \@tempswafalse \fi
```

Restart of line numbers / different page!?

```
         \ifx\@EN@blemp\@EN@elemp \else \@tempswafalse \fi
         \if@tempswa
           \sameline{\@EN@plref\@EN@blemp\@EN@bleml}%
         \else
           \@EN@lastline@z@
           \differentlines{\@EN@plref\@EN@blemp\@EN@bleml}%
                          {\ifx\@EN@blemp\@EN@elemp
                             \@EN@eleml
                           \else
                             \pageandline\@EN@elemp\@EN@eleml
                           \fi}%
           \@EN@lastline@z@
         \fi}%
      \lemmafmt{{\@EN@lemmaexpands#3}}%
      \notefmt{#4}%
    }%
  }%
```

A copy \EFN@@appentry for the case that \@EN@appentry is redefined by \FNLN@text:

```
\let \EFN@@appentry \@EN@appentry
```

2.3.3 Basic Changes

We patch the patch of \@footnotetext again ... \EFN@@text stores fnlineno's variant \FNLN@text of LaTeX's \@footnotetext:

```
\let \EFN@@text \FNLN@text
```

A hook \EFN@annot@hook will collect and handle annotations to the footnotes of a page. It will be changed globally:

```
\global\let \EFN@annot@hook \@empty
```

\EFN@move{⟨*id*⟩}{⟨*note-fam*⟩}{⟨*lemma-tag*⟩}{⟨*note*⟩} will move an annotation into the (next ...) hook:

```
\def \EFN@move #1#2#3#4{%
    \g@addto@macro\EFN@annot@hook{\EFN@annot{#1}{#2}{#3}{#4}}}
```

The new \FNLN@text will redefine \@EN@appentry in order to redirect the annotation:

```
\renewcommand\FNLN@text[1]{\EFN@@text{\let\@EN@appentry\EFN@move #1}}
```

2.4 Inserting Annotations to Footnotes

2.4.1 Strategy

Version v0.2 of edfnotes.sty aims at proper **sorting** of the annotations, in the sense that in the critical apparatus, notes on *main text* lines appear before all notes to *footnote* lines.

David Kastrup's bigfoot and perpage packages seem to deal with this task (however, the documentation doesn't help much in making use of it); on the other hand, lineno.sty offers its own nice tools ...

Our idea is inserting all the *annotations to footnotes* right after the *last main text line of the page*—this way they should appear *after* all annotations to main text lines. (This idea grew with v0.2, and with v0.3 it became clear this hook can be used for passing certain annotations to the next page.)

For version v0.3, this is refined a little. The hook will not actually \insert all annotations, it will rather choose some for immediate insertion and collect the remaining annotations in the hook for the next page.

We use \MakeLineNo from lineno.sty for hooking in here (you may find the context relevant here in fnlineno.pdf). \MakeLineNo—in "pagewise" mode that we assume here—has access to the absolute number of the last numbered line of the current page, involving lineno.sty's \testLastNumberedPage. We modify the latter so that it records that number as \theLastLineNumber.

2.4.2 Deciding

`\EFN@run@annot@hook` first expands `\EFN@annot@hook`, then empties it, and while the version for the recent page runs, the version of `\EFN@annot@hook` for the *next* page may be built, containing annotations to footnote passages on later pages:

```
\def \EFN@run@annot@hook {%
    \expandafter \global \expandafter \let
        \expandafter \EFN@annot@hook \expandafter \@empty
    \EFN@annot@hook}
```

`\EFN@annot@hook` (unless empty) is a list of commands

`\EFN@annot{`⟨*id*⟩`}{`⟨*note-fam*⟩`}{`⟨*lemma-tag*⟩`}{`⟨*note*⟩`}`

that were issued either by `\@EN@putdown` or by the previous version of `\EFN@run@annot@hook` using `\EFN@move`. `\EFN@annot` uses ⟨*id*⟩ to determine the page number where the footnote passage the annotation refers to starts. There are *three* cases deciding about immediate `\insert`ion vs. postponing. The annotation is `\insert`ed immediately if *either* ⟨*id*⟩ has not been used in the previous run (it contains an annotation number exceeding the earlier number of annotations—minus one) *or* if the annotation was placed on the "current" page in the run before. In these cases, the four arguments of `\EFN@annot` are run by `\EFN@@appentry`. *Otherwise* the annotation is deferred by running the four arguments with `\EFN@move`.

```
\def \EFN@annot #1{%
  \let \EFN@next \EFN@@appentry
```

I.e., immediate `\insert`ion is the "default." `undefined` in the next line means there is no placement information from the previous run (stored as `\r@`⟨*id*⟩). If there is, it is extracted by `\@EN@extract`, `\@EN@blemp` will carry the "public" (relative) page number.

```
  \@ifundefined{r@#1}\relax{%
    \@EN@xpxpxp \@EN@extract \csname r@#1\endcsname
      \relax \relax \relax \relax %% somewhat lazier than ednotes
```

If the page numbers don't match, we `move`. With v0.6 however, we don't `move` when we have "missed" the (according to `.aux`) matching page. This happens when the passage we are referring to moves to later pages.

```
%     \ifnum \@EN@blemp=\c@page     %% TODO \ifx\relax!? \c@page OK?
%     \else
    \ifnum \@EN@blemp>\c@page     %% v0.6
      \let \EFN@next \EFN@move
    \fi }%
  \EFN@next{#1}%
}
```

2.4.3 Normal Insertion

The "normal" way of inserting annotations to footnotes is a call from the extended `\MakeLineNo` involving testing whether the line is the last numbered one on the page. It seems not to have been easily accessible with lineno.sty, we are extending the latter's `\testLastNumberedPage` so it stores that number as `\theLastLineNumber`. Indeed, calculating the "pagewise" line number in `\MakeLineNo` invokes a series of tests that finally calls

`\testLastNumberedPage{`⟨*int*⟩`}`

where ⟨*int*⟩ is the least "last" line number of a page with numbered lines that is greater than or equal to the current absolute line number.

```
\def\testLastNumberedPage#1{\ifnum#1<\c@linenumber
      \let\firstLN\@gobble
    \fi
    %% both tests new 2011/01/15,
    %% not sure about efficiency TODO:
%     \if@FNLN@sw@ \else
% <- don't run with \getfoot...! 2011/01/19 ->
    \ifx\c@linenumber\c@pagewiselinenumber
      \ifnum\theLastLineNumber=#1\relax \else
        \gdef\theLastLineNumber{#1}%        %% mod. 2011/01/07!
    \fi \fi }
```

lineno's `\MakeLineNo` is extended using that `add` macro from ednotes. It will test if the line just numbered is a main text line and the last numbered one on the page, and only then call the `\insert`s from manyfoot for the critical apparatus.—With v0.5, we actually extend `\stepLineNumber` that `\MakeLineNo` invokes. Appending `\insert`ions to `\MakeLineNo` creates a strong risk that it is executed on the *next* page. (So this requires a certain version of lineno.sty, and the latter might better provide an "official" hook. *Prepending* to `\MakeLineNo` in v0.34 failed because `\theLastLineNumber` may be wrong there.)

```
\@EN@addtomacro \stepLineNumber {%
  \if@FNLN@sw@ \else
    \ifx\EFN@annot@hook\@empty \else    %% 2011/01/14 efficient!? TODO
      \advance\c@linenumber\m@ne        %% local to \output 2011/01/16
      \ifnum\theLastLineNumber=\c@pagewiselinenumber
        \EFN@run@annot@hook
  \fi \fi \fi }
```

2.4.4 Forced Insertion

1. At an **initial run** of the document (no `.aux` present), there isn't any information about line number ranges for pages. We then `\insert` everything immediately so that an initial run at least produces page breaks that are useful for ordering annotations in the next run. This is achieved by choosing a default

definition for `\theLastLineNumber` that trivially renders the `\ifnum` test in `\MakeLineNo \iftrue`.

```
\@ifdefinable \theLastLineNumber {%
          \let \theLastLineNumber \c@pagewiselinenumber}
```

2. Another problem may occur with the **very last line** of an edition. When between two LaTeX runs the number of main text lines is reduced (e.g., one has removed wrong text or has improved line breaking by a manual change of hyphenation), the annotations for the last page may get lost because the line number that the insertions wait for doesn't occur any more. They will then get lost, and their `\newlabel` entries in the `.aux` file will be missing, perhaps resulting in other errors. We provide a command `\ForceFootnoteAnnotations` for manual use by authors or for being added to certain hooks. It might be added to `\nolinenumbers` and `\endlinenumbers`, but I am not sure ...

v0.5 allows a much cleaner implementation of `\ForceFootnoteAnnotations` than we had in v0.35 and v0.4:

```
\newcommand* \ForceFootnoteAnnotations {{%
    \let\EFN@annot\EFN@@appentry
    \EFN@run@annot@hook}}
```

3. Finally it may occur that a **chapter ends** with a very **long footnote** with annotations to all of its pages, and the editor decides that the footnotes should fill the final pages of the chapter without main text on their pages. Then the annotations have problem with our approach of v0.3. A similar problem could be that a short end of the last footnote of a chapter gets remarks of several pages that the editor doesn't want to see in the following chapter (e.g., when there isn't any following chapter).

We therefore provide a command `\clearfootnoteannotations` that creates dummy page entries for those final pages. Of course, these dummy page entries must not be numbered like lines ...

The first version of the macro worked fine with a two-page footnote, but crashed as soon as a main text line was removed. (Infinitly many pages are created then ...) This is just the problem that was addressed by `\ForceFootnoteAnnotations`. But it would be very wrong to use `\ForceFootnoteAnnotations` for the present problem, because this would move the annotations for the next footnote-only pages to the next page with numbered main text, or to the end of the entire document.

The problem has then been solved by inserting `\EFN@run@annot@hook`; however, in reasoning about it, about former failures, about the `\par`, it seems to become clear that this is a solution for Christian's `Test13.tex` as of 2011/01/16 after a footnote that closes a paragraph and a main text page at the same time—only, while ...

```
\newcommand* \clearfootnoteannotations {%
  \par          %% TODO!? the whole only via \MakeLineNo!?
```

```
                %%        doesn't work without \par
  \EFN@run@annot@hook
    \loop \ifx\EFN@annot@hook\@empty \else
      \pagebreak          %% TODO!? \newpage? \clearpage?
                          %%       leave to user before?
      \write\m@ne{}\vbox{}%% TODO!? from \clearpage
      \EFN@run@annot@hook
    \repeat
  %% TODO: \clearpage!? may come from \chapter
}
```

TODO: What if pages are removed? What if annotations are inserted or removed? So far, ...

2.5 ednotes' \newlabel variant

ednotes changes LaTeX's `\newlabel` so that three `\newlabel` entries in the `.aux` file build only *one* macro containing the information needed for typesetting *one* annotation in the apparatus. Both that building mechanism and the change testing at the final run of the new `.aux` file don't work with edfnotes.sty. To repair this, I needed much time to understand those mechanisms again. I found that they were quite redundant and now am re-implementing them almost entirely.

2.5.1 Suppressing the "\get..." Commands

lineno must suppress expansion of `\getpagewiselinenumber` when the `.aux` file is read. With fnlineno.sty, `\getfootnotelinenumber` must be prevented from expansion as well. Both macros are activated at `\begin{document}` only when reading the `.aux` has been finished.

With fnlineno.sty, both macros are accessed as `\getwiselinenumber`⟨*cs*⟩, so we just need to switch `\getwiselinenumber`:

```
% \let\EFN@getlineno\getwiselinenumber
% \let\getwiselinenumber\@gobble
% \AtBeginDocument{\let\getwiselinenumber\EFN@getlineno}
```

On the other hand, ednotes deals with `\getpagewiselinenumbers` already, so we only ... saves one expansion for each footnote line!

```
\let \EFN@getfnlineno \getfootnotelinenumber
\let \getfootnotelinenumber \relax
\AtBeginDocument{\let \getfootnotelinenumber \EFN@getfnlineno}
```

By analogy to ednotes' `\AtEnd...`:

```
\AtEndDocument{\let\getfootnotelinenumber\relax}
```

2.5.2 `\newlabel` Building Info Macros

For version v0.3 of edfnotes, it turns out that ednotes' (v1.1–1.3) mechanism for building the note info macros `\r@EN@l`⟨*id*⟩ assumes that

`\newlabel{-EN@l`⟨*id*⟩`}{`⟨*start-lemma-place*⟩`}`

will appear before

`\newlabel{+EN@l`⟨*id*⟩`}{`⟨*end-lemma-place*⟩`}`

and

`\newlabel{EN@l`⟨*id*⟩`}{`⟨*note-place*⟩`}`

in the `.aux` file. With edfnotes v0.3, an annotation to the second page of a long footnote will be `\insert`ed on the previous page in a "first" run (no `.aux` present) due to the missing placement information. ednotes' `\newlabel` mechanism then looses the note placement information, and a "missing number" error occurs.

To fix this, we modify ednotes' `\@EN@addtolabeldef` so that at the first step of the recursive building of `\r@EN@l`⟨*number*⟩, the latter is not used as input (as a part). Then the tricky `\@gobble` in the first case of ednotes' redefinition of `\newlabel` (see the recent presentation of the code—currently [2011/01/10] `ednotesc.pdf`) can be replaced by the obvious empty argument.

With the former implementation in ednotes, I had already wondered if I couldn't postpone reading the second argument of `\newlabel{`⟨*label*⟩`}{`⟨*info*⟩`}`. Now this is what I am doing indeed. ednotes stores the original `\newlabel` as `\@EN@newlabel`, and this will be called for the default treatment of ⟨*label*⟩ and ⟨*info*⟩: For re-implementing the `\newlabel` mechanisms, first we don't use LaTeX's `\in@`, but our own test whether `EN@l` is part of a `\newlabel` ⟨*label*⟩:

```
\renewcommand* \newlabel [1] {%
    \def\@EN@next{\@EN@newlabel{#1}}%
    \@EN@test@noteid #1\@empty EN@l\@empty\@nil
    \@EN@next}
```

Here are macros that `\newlabel` invokes:

`\@EN@labels` chooses a "name space" for labels. The intention is that an `.aux` entry `\newlabel{`⟨*label*⟩`}{`⟨*info*⟩`}` contributes note placement information *iff* ⟨*label*⟩ has form ⟨*sign*⟩`EN@l`⟨*digits*⟩ where ⟨*sign*⟩ is nothing, `-`, or `+`. However, we won't really (soon) check if we have digits indeed ...

```
\newcommand* \@EN@labels {EN@l}
```

The next line defines a macro `\EN@l` calling `\@EN@addtolabeldef` one way ...

```
\@namedef {\@EN@labels}{\@EN@addtolabeldef{{\@EN@blemp}}{}}
```

The next two lines define a macro `\-EN@l` calling `\@EN@addtolabeldef` another way ...

```
\@namedef{-\@EN@labels}{%
    \@EN@addtolabeldef{}{{\@EN@blem1}{\@EN@blemp}}}
```

The next two lines render `\+EN@l` an alias for `\-EN@l` ...

```
\expandafter \let \csname +\@EN@labels \expandafter\endcsname
    \csname -\@EN@labels\endcsname
```

`\@EN@test@noteid`⟨*split1*⟩`EN@l`⟨*split2*⟩`\@empty`⟨*split3*⟩`\@nil` is meant for application as `\@EN@test@noteid`⟨*label*⟩`\@empty␣EN@l\@empty\@nil`. (I *think* other control sequences than `\@empty` would do as well.) When the macro considers ⟨*label*⟩ belonging to the note info system, `\@tempa` equals one out of `\EN@l`, `\-EN@l`, and `\+EN@l`, and `\@EN@tempa` expands to `\r@`⟨*split1*⟩`EN@l`⟨*split2*⟩.

```
\def \@EN@test@noteid #1EN@l#2\@empty#3\@nil{%
    \ifx\\#2\\\else
```

This should mean that `#2` is *not* empty (I *think* the control sequence `\\` can't appear in a label ...), and this is the case *iff* ⟨*label*⟩ contains `EN@l` and something more to the *right*. `#1` (= ⟨*split1*⟩) then is the part of ⟨*label*⟩ to the *left* of that occurrence of `EN@l`. Further conditions for ⟨*label*⟩ belonging to a note are (*i*) `#1` is empty, `-`, or `+`—we check this by comparing `\`⟨*split1*⟩`EN@l` to `\EN@l`, `\-EN@l`, and `\+EN@l`—, and (*ii*) `#2` is some ⟨*digits*⟩—that we won't check (soon) ...

```
        \expandafter \let \expandafter \@tempa
                \csname #1\@EN@labels\endcsname
        \ifx\@tempa\relax \else
            \let \@EN@next \@tempa
```

This should be the same `\@EN@tempa` as `\@EN@extractcslp` from ednotes v1.3 produced it:

```
            \edef \@EN@tempa {\expandafter\noexpand
                \csname r@\@EN@labels#2\endcsname}%
          \expandafter
```

The remaining part of former `\@EN@extractcslp` should be called by the new version of `\@EN@addtolabeldef`. The following hook will be used in testing for cross-reference changes:

```
            \@EN@testnote@hook
        \fi
    \fi
}
```

ednotes' `\@EN@addtolabeldef` even gets a new syntax:

`\@EN@addtolabeldef{`⟨*prepend*⟩`}{`⟨*append*⟩`}{`⟨*line/page*⟩`}`

```
\renewcommand* \@EN@addtolabeldef [3] {%
    \@EN@extract@lp#3\@nil
    \expandafter \xdef \@EN@tempa {%
        #1%
       \expandafter
        \ifx\@EN@tempa\relax \else \@EN@tempa \fi
        #2}%
    \@EN@testnote@h@@k}
```

... considering ednotes' \protected... irrelevant. ednotes' \@EN@extractcslp is replaced by \@EN@extract@lp⟨*line/page*⟩\@nil:

```
\let \@EN@extractcslp \@undefined   %% must be renamed:
\def \@EN@extract@lp #1#2#3\@nil {%
        \def\@EN@bleml{#1}\def\@EN@blemp{#2}}
```

The \@EN@testnote@hook and \@EN@testnote@h@@k will only be used at \end{document}:

```
\let \@EN@testnote@hook \relax
\let \@EN@testnote@h@@k \relax
```

2.5.3 \newlabel Testing Cross-reference Changes

We won't use \@EN@testlabel and must override ednotes' command to use it in the testing run of the .aux (maybe we should care that ednotes has been loaded before!?). Instead, the hook \@EN@testnote@hook will be activated. For the overriding, we need to store our present redefinition of \newlabel. (When our re-implementation of \newlabel is merged into ednotes, there simply shouldn't any change of \newlabel at \end{document}.)

```
\@onlypreamble\@EN@testlabel
\let \EFN@newlabel \newlabel
\AtEndDocument{%
    \let \newlabel \EFN@newlabel
    \def \@EN@testnote@hook {%
```

From ednotes' \@EN@testlabel:

```
      \expandafter
        \ifx\@EN@tempa\relax
```

2011/01/13: This means that the note info macro that \@EN@tempa expands to—call it ⟨*id*⟩ for the discussion—has been used in the present run of the document, but not in the run before (provided ...—see below). So certainly this "label has changed," and

```
          \@tempswatrue
```

should issue LaTeX's according final warning. Moreover, the ⟨*info*⟩ argument of the current \newlabel entry should now simply be ignored (and this way the same \@EN@tempa will expand to a \relax alias next time again):

```
          \let\@EN@next\@gobble
        \else
```

This means that the note info macro ⟨*id*⟩ has been used both in the present run and in the run before, so we want to find out whether the info changed or not. We will find three \newlabel entries for the same annotation and must build its new info macro—the string ⟨*new-info*⟩ that ⟨*id*⟩ will expand to after reading the

`.aux` in the next document run—, we must store the old one—⟨*id*⟩ still expands to ⟨*old-info*⟩—for comparing, we must be able to detect if we have collected all the three entries that we need for ⟨*new-info*⟩, and we must not increase the number of control sequences much. In ednotes we used *one* separate "testing list" for all the info macros to be tested. The situation is now quite different and we try a new way: We collect all the relevant informations in that info macro ⟨*id*⟩ and take them apart when we have all of them.

We are building ⟨*new-info*⟩, at first it is empty. We will insert

`\@EN@record`⟨*count*⟩`{`⟨*old-info*⟩`}`,

together with the contribution from the first `\newlabel`. Let us see if the ⟨*new-info*⟩ from the current ⟨*id*⟩ already contains such a `\@EN@record`, using `\@EN@test@record`:

```
177             \@EN@xpxpxp \@EN@test@record \@EN@tempa \@EN@record0\@nil
178           \fi}%
179   }
180   \def \@EN@test@record #1\@EN@record#2#3\@nil{%
181       \if #20%
```

`record` is missing (this conclusion is justified by the rest of the macro), this is the first contribution to a note info macro. We are preparing the insertion that `\@EN@addtolabeldef` will execute—`record␣1{`⟨*old-info*⟩`}`:

```
182           \let\@EN@record\relax
183           \expandafter\edef\@EN@tempa{\@EN@record 1{\@EN@tempa}}%
184       \else
185           \if #21%
```

Here we have found a *second* contribution.

```
186               \def\@EN@record##1{\noexpand\@EN@record 2}%
187           \else
```

This is the case where the *third* `\newlabel` for the same annotation has been encountered. When its ⟨*info*⟩ argument has been processed, we can compare the resulting ⟨*new-info*⟩ to the ⟨*old-info*⟩ that we are now putting from `#3` into `\@tempa`:

```
188               \@EN@xpxpxp \@EN@extract@record \@EN@tempa \@nil
```

Now `\@tempa` should expand to ⟨*old-info*⟩.

```
189               \let\@EN@record\@gobbletwo
190           \fi
191       \fi
192   }
```

`\@EN@extract@record` prepares comparison of ⟨*old-info*⟩ and ⟨*new-info*⟩:

```
193   \def \@EN@extract@record #1\@EN@record 2#2#3\@nil {\def\@tempa{#2}}
```

This is the active `\@EN@testnote@h@@k` providing testing and warning about annotation placement changes:

```
\AtEndDocument{%
    \def \@EN@testnote@h@@k {%
```

When another reference has changed, we may omit this info (probably different to ednotes v1.3).

```
      \if@tempswa \else
        \ifx\@EN@record\@gobbletwo
         \expandafter
          \ifx \@EN@tempa \@tempa \else
            \@tempswatrue   %% for LaTeX's warning TODO!?
            \@EN@linechange@warning
          \fi
        \fi
    \fi}%
}
\newcommand* \@EN@linechange@warning{%
```

Now copying ednotes v1.3 (with different code line breaks):

```
      \expandafter\@EN@takepagesto\@tempa\@tempa
      \@EN@xpxpxp \@EN@takepagesto\@EN@tempa\@EN@tempa
```

The message may come when line numbers have changed, not page numbers. This may puzzle some users, but should be harmless.

```
      \typeout{^^JPackage `ednotes.sty':^^J%
          *** A lemma or note moved. Page numbers \@tempa\space turned
          into \@EN@tempa.^^J*** Rerun and watch whether they come to
          rest and references get right.^^J*** If they don't, use
          \string\pagebreak\space or \string\warningpagebreak\space
          to force^^J*** the earlier of oscillating page breaks.^^J***
          (Cf. package documentation on `oscillating page breaks'.)}%
    \let\newlabel\@EN@newlabel
    \def\@newl@bel{\@gobblefour\relax}% Cf. `lblchng1.sty'.
}
```

`\@EN@prepare@fromtl` is obsolete:

```
\@onlypreamble \@EN@prepare@fromtl
```

2.5.4 Finish Typesetting before Testing

When the numbering macros are disabled for testing before finishing typesetting, absolute line numbers instead of relative ones are printed. The disabling commands are issued using `\AtEndDocument` which appends the commands to the `\@enddocumenthook`. We *prepend* a `\par` token to this hook so typesetting the numbered paragraph happens early enough, likewise we may force final annotations whose "last" line numbers have been lost:

```
\toks@\expandafter{\@enddocumenthook}
```

```
\edef\@enddocumenthook{%
    \noexpand\par
    \noexpand\ForceFootnoteAnnotations
    \the\toks@}
```

We might have used Heiko Oberdiek's atveryend instead of `\AtEndDocument`; and I thought of adding another `\clearpage`, but this may conflict with other packages. These matters are described comprehensively in the documentation of the lastpage package.

2.6 Leaving the Package File

```
\endinput
```

2.7 VERSION HISTORY

```
v0.1    2010/01/01  starting, trying to change \FootnotetextA
        2011/01/01  discovered \@EN@hookfn, exploited
        2011/01/02  another task: ednotes' \newlabel
        2011/01/03  disabling \getfoot... for .aux already;
                    added \par to \enddocument;
                    extended 'Limitations', add-macros -> cs;
                    removed debugging code
        TO CHRISTIAN as part of EDFN-r0.4
v0.2    2011/01/04  simplified hook stuff, moving to fnlineno
        2011/01/07  <- moving back, insert annotations at
                    last line of page, restructuring;
                    \theLastLineNumber
        TO CHRISTIAN as part of EDFN-r0.5 night to 2011/01/08
v0.3    2011/01/08  concept
        2011/01/09  samepage -> page@end;
                    details for concept, implementation
        2011/01/10  correction of ednotes' \newlabel
v0.31   2011/01/10  JUST STORING: first approach at
                    repairing \newlabel
v0.32   2011/01/11  restructuring and explanation for
                    re-implementation of \newlabel,
                    implementation of building
        JUST STORING -- seems to work so far, but there is
                    old code that just is overridden
v0.33   2011/01/12  corr. doc. mistakes on test@noteid;
                    hard work at both versions of \newlabel ...
        2011/01/13  untidy, but seems to work perfectly
v0.34   2011/01/14  rough and ready fix: \insert at *start*
                    of \MakeLineNo
        TO CHRISTIAN without any other files
v0.35   2011/01/14  acc. Christian debugging
        2011/01/15  assignment of \theLastLineNumber conditional,
                    another change of \MakeLineNo: insertion hook
```

```
259                     *after* \makeLineNumber as in v0.33,
260                     \c@linenumber stepped back
261          2011/01/16 rm. some \show...; \if@FNLN@sw@ replaces \ifx,
262                     \EFN@insert@try, \ForceFootnoteAnnotations;
263                     \par before \newlabel tests, note on
264                     \clearpage and related packages;
265                     restructured inserting section;
266                     tidied \newlabel section
267          TO CHRISTIAN as part of EDFN r0.55
268   v0.4   2011/01/17 \EFN@page@end -> annot, \fi \fi ...;
269                     \EFN@insert@try merged into \MakeLineNo again,
270                     new \EFN@run@annot@hook, rm. \EFN@trivialize@last;
271                     \paragraph's in "Forced", compl. doc.
272                     \EFN@extract@record; \clearfootnoteannotations
273   v0.5   2011/01/18 re-impl. \@EN@putdown etc. entirely, much new doc.
274          2011/01/19 \testLastNumberedPage uses \ifx\c@linenumber...;
275                     appended hook run to \stepLineNumber;
276                     \ForceFootnoteAnnotations re-implemented;
277                     doc. fixes \paragraph{3.}
278   v0.6   2011/02/02 \EFN@move in \EFN@annot only *before* lemma page,
279                     copyright updated, DFG reduced, user command*s*
280   v0.6a  2011/02/10 "Known Issues"
281   v0.6b  2011/02/14 modifications for Christian's 2011/02/11;
282                     mention 'PdUsample.pdf'; history *sub*section;
283                     'tamefloats' link corrected
284          2011/02/16 install: required files, \pagebreaks
285
```

3 Example

The file `PdUsample.pdf` shows results with a very long (original, edited) footnote in Bernard Bolzano's *Paradoxien des Unendlichen*, with critical annotations by Christian Tapp.