eledmac

Typeset scholarly editions with \LaTeX *

Maïeul Rouquette

based on the original ledmac by
Peter Wilson
Herries Press

which was based on the original EDMAC, TABMAC and EDSTANZA by
John Lavagnino, Dominik Wujastyk, Herbert Breger and Wayne Sullivan.

Abstract

EDMAC, a set of Plain \TeX macros, was made at the beginning of 90’s for
typesetting critical editions in the traditional way, i.e., similar to the Oxford
Classical Texts, Teubner, Arden Shakespeare and other series. A separate
set of Plain \TeX macros, TABMAC, provides for tabular material. Another
set of Plain \TeX macros, EDSTANZA, assists in typesetting verse.

The eledmac package makes the EDMAC, TABMAC and EDSTANZA facilities
available to authors who would prefer to use \LaTeX. The principal functions
provided by the package are marginal line numbering and multiple series of
foot- and endnotes keyed to line numbers.

In addition to the EDMAC, TABMAC and EDSTANZA functions the package also
provides for index entries keyed to both page and line numbers. Multiple
series of the familiar numbered footnotes are also available.

Other \LaTeX packages for critical editions include EDNOTES, and poemscol
for poetical works.

eledmac provides many tools and options. Normally, they are all docu-
mented in this file. Also provided is a help folder, "examples". The folder
contains additional examples (although not for all cases). Example starting
by “1-” are for basic uses, those starting by “2-” are for advanced uses.

To report bugs or request a new feature, please go to ledmac GitHub
page and click on "New Issue": https://github.com/maieul/ledmac/
issues/. You must create an account on github.com to access my page
(maieul/ledmac). GitHub accounts are free for open-source users. You can
post messages in English or in French (preferred).

You can subscribe to the eledmac mail list in:
http://geekographie.maieul.net/146

\*This file (eledmac.dtx) has version number v1.23.2, last revised 2015/05/29.
\^maieul at maieul dot net
# Contents

<table>
<thead>
<tr>
<th>1 Introduction</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1 Overview</td>
<td>7</td>
</tr>
<tr>
<td>1.2 History</td>
<td>9</td>
</tr>
<tr>
<td>1.2.1 EDMAC</td>
<td>9</td>
</tr>
<tr>
<td>1.2.2 eledmac</td>
<td>10</td>
</tr>
<tr>
<td>1.2.3 List of works edited with (e)ledmac</td>
<td>11</td>
</tr>
</tbody>
</table>

| 2 The eledmac package | 11 |

| 3 Options | 11 |

| 4 Text lines and paragraphs numbering | 12 |
| 4.1 Text lines numbering | 12 |
| 4.2 Paragraphs | 13 |
| 4.2.1 Basis | 13 |
| 4.2.2 Content before specific \pstart and after \pend | 13 |
| 4.2.3 Content before every \pstart and after every \pend | 14 |
| 4.2.4 Producing automatically \pstart...\pend | 14 |
| 4.2.5 Numbering paragraphs (\pstart) | 14 |
| 4.2.6 Languages written in Right to Left | 15 |
| 4.2.7 Memory limits | 15 |
| 4.3 Lineation commands | 16 |
| 4.3.1 Disabling lineation | 16 |
| 4.3.2 Setting lineation start and step | 16 |
| 4.3.3 Setting lineation reset | 16 |
| 4.3.4 Setting line number margin | 16 |
| 4.3.5 Other settings | 17 |
| 4.4Changing the line numbers | 17 |

| 5 The apparatus | 18 |
| 5.1 Commands | 18 |
| 5.1.1 The lemma | 18 |
| 5.1.2 Footnotes | 19 |
| 5.1.3 Endnotes | 20 |
| 5.1.4 Paragraph in critical apparatus | 20 |
| 5.2 Disambiguation of identical words in the apparatus | 22 |
| 5.2.1 Basic use | 22 |
| 5.2.2 Note about input encoding with UTF-8 processor | 22 |
| 5.2.3 Use with \lemma command | 23 |
| 5.2.4 Customizing | 24 |
| 5.3 Alternate footnote formatting | 24 |
| 5.4 Display options | 25 |
| 5.4.1 Control line number printing | 25 |
| 5.4.2 Separator between the lemma and the note | 28 |
## Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.4.3 Font style</td>
<td>29</td>
</tr>
<tr>
<td>5.4.4 Font of the lemma</td>
<td>29</td>
</tr>
<tr>
<td>5.4.5 Styles of notes content</td>
<td>29</td>
</tr>
<tr>
<td>5.4.6 Arbitrary code at the beginning of notes</td>
<td>30</td>
</tr>
<tr>
<td>5.4.7 Options for footnotes in columns</td>
<td>30</td>
</tr>
<tr>
<td>5.4.8 Options for paragraphed footnotes</td>
<td>31</td>
</tr>
<tr>
<td>5.4.9 Options for block of notes</td>
<td>31</td>
</tr>
<tr>
<td>5.5 Page layout</td>
<td>32</td>
</tr>
<tr>
<td>5.5.1 Endnotes in one paragraph</td>
<td>32</td>
</tr>
<tr>
<td>5.6 Fonts</td>
<td>32</td>
</tr>
<tr>
<td>5.7 Changing series</td>
<td>34</td>
</tr>
<tr>
<td>5.7.1 Create a new series</td>
<td>34</td>
</tr>
<tr>
<td>5.7.2 Delete series</td>
<td>34</td>
</tr>
<tr>
<td>5.7.3 Series order</td>
<td>34</td>
</tr>
<tr>
<td>6 Verse</td>
<td>34</td>
</tr>
<tr>
<td>6.1 Repeating stanza indents</td>
<td>35</td>
</tr>
<tr>
<td>6.2 Manual stanza indent</td>
<td>35</td>
</tr>
<tr>
<td>6.3 Stanza breaking</td>
<td>36</td>
</tr>
<tr>
<td>6.4 Hanging symbol</td>
<td>36</td>
</tr>
<tr>
<td>6.5 Long verse and page break</td>
<td>37</td>
</tr>
<tr>
<td>6.6 Various tools</td>
<td>37</td>
</tr>
<tr>
<td>6.7 Hanging symbol</td>
<td>37</td>
</tr>
<tr>
<td>6.8 Text before/after verses</td>
<td>37</td>
</tr>
<tr>
<td>7 Grouping</td>
<td>38</td>
</tr>
<tr>
<td>8 Crop marks</td>
<td>38</td>
</tr>
<tr>
<td>9 Cross referencing</td>
<td>38</td>
</tr>
<tr>
<td>9.1 Basic use</td>
<td>39</td>
</tr>
<tr>
<td>9.2 Normal LATEX cross-referencing</td>
<td>40</td>
</tr>
<tr>
<td>9.3 References to lines commented in the apparatus</td>
<td>40</td>
</tr>
<tr>
<td>10 Side notes</td>
<td>41</td>
</tr>
<tr>
<td>11 Familiar footnotes</td>
<td>42</td>
</tr>
<tr>
<td>11.1 Position of the familiar footnotes</td>
<td>43</td>
</tr>
<tr>
<td>12 Indexing</td>
<td>43</td>
</tr>
<tr>
<td>12.1 Using xindy</td>
<td>44</td>
</tr>
<tr>
<td>13 Tabular material</td>
<td>45</td>
</tr>
<tr>
<td>14 Sectioning commands</td>
<td>48</td>
</tr>
<tr>
<td>14.1 Sectioning commands without line numbers or critical notes</td>
<td>48</td>
</tr>
<tr>
<td>14.2 Sectioning commands with line numbering and critical notes</td>
<td>48</td>
</tr>
</tbody>
</table>
Contents

15 Quotation environments 50
16 Page breaks 50
17 Miscellaneous 51
  17.1 Known and suspected limitations 51
  17.2 Use with other packages 52
  17.3 Parallel typesetting 53
18 Implementation overview 54
19 Preliminaries 54
  19.1 Package options 55
  19.2 Loading packages 56
  19.3 Boolean flags 56
  19.4 Messages 57
  19.5 Gobbling 61
  19.6 Miscellaneous commands 61
20 Sectioning commands 61
21 Line counting 65
  21.1 Choosing the system of lineation 65
  21.2 List macros 70
  21.3 Line-number counters and lists 71
  21.4 Reading the line-list file 75
  21.5 Commands within the line-list file 77
  21.6 Writing to the line-list file 85
22 Marking text for notes 89
  22.1 \edtext (and \critext) itself 90
  22.2 Substitute lemma 96
  22.3 Substitute line numbers 97
  22.4 Lemma disambiguation 97
23 Paragraph decomposition and reassembly 103
  23.1 Boxes, counters, \pstart and \pend 103
  23.2 Processing one line 107
  23.3 Line and page number computation 110
24 Line number printing 113
  24.1 Pstart number printing in side 117
  24.2 Add insertions to the vertical list 118
  24.3 Penalties 119
  24.4 Printing leftover notes 120
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>34 Display</td>
<td>173</td>
</tr>
<tr>
<td>34.1 Change series order</td>
<td>173</td>
</tr>
<tr>
<td>34.2 Test series order</td>
<td>173</td>
</tr>
<tr>
<td>34.3 Options</td>
<td>174</td>
</tr>
<tr>
<td>34.3.1 Tools to set options</td>
<td>174</td>
</tr>
<tr>
<td>34.3.2 Tools to generate options commands</td>
<td>175</td>
</tr>
<tr>
<td>34.3.3 Options for critical notes</td>
<td>176</td>
</tr>
<tr>
<td>34.3.4 Options for familiar notes</td>
<td>177</td>
</tr>
<tr>
<td>34.3.5 Common options to critical and familiar footnotes</td>
<td>178</td>
</tr>
<tr>
<td>34.3.6 Options for endnotes</td>
<td>178</td>
</tr>
<tr>
<td>34.4 Old commands, kept for backward compatibility</td>
<td>178</td>
</tr>
<tr>
<td>34.5 Hooks for a particular footnote</td>
<td>179</td>
</tr>
<tr>
<td>34.6 Alias</td>
<td>179</td>
</tr>
<tr>
<td>35 Line number printing</td>
<td>179</td>
</tr>
<tr>
<td>36 Output routine</td>
<td>182</td>
</tr>
<tr>
<td>37 Cross referencing</td>
<td>189</td>
</tr>
<tr>
<td>38 Side notes</td>
<td>196</td>
</tr>
<tr>
<td>39 Minipages and such</td>
<td>202</td>
</tr>
<tr>
<td>40 Indexing</td>
<td>206</td>
</tr>
<tr>
<td>40.1 Memoir compatibility</td>
<td>209</td>
</tr>
<tr>
<td>40.2 Normal setting</td>
<td>211</td>
</tr>
<tr>
<td>40.3 Choose the right variant</td>
<td>212</td>
</tr>
<tr>
<td>40.4 hyperref compatibility</td>
<td>212</td>
</tr>
<tr>
<td>41 Macro as environment</td>
<td>214</td>
</tr>
<tr>
<td>42 Verse</td>
<td>218</td>
</tr>
<tr>
<td>43 Arrays and tables</td>
<td>222</td>
</tr>
<tr>
<td>44 Section’s title commands</td>
<td>242</td>
</tr>
<tr>
<td>44.1 Deprecated commands</td>
<td>242</td>
</tr>
<tr>
<td>44.2 New commands : \eledxxx</td>
<td>244</td>
</tr>
<tr>
<td>45 Page breaking or no page breaking depending of specific lines</td>
<td>254</td>
</tr>
<tr>
<td>46 Long verse: prevents being separated by a page break</td>
<td>256</td>
</tr>
<tr>
<td>47 The End</td>
<td>256</td>
</tr>
</tbody>
</table>
1 Introduction

The EDMAC macros \cite{LW90} for typesetting critical editions of texts have been available for use with TeX since 90’s. Since EDMAC was introduced there has been a small but constant demand for a version of EDMAC that could be used with LaTeX. The eledmac package is an attempt to satisfy that request.

eledmac would not have been possible without the amazing work by John Lavagnino and Dominik Wujastyk, the original authors of EDMAC. I, Peter Wilson, am very grateful for their encouragement and permission to use EDMAC as a base. The majority of both the code and this manual are by these two. The tabular material is based on the TABMAC code \cite{Bre96}, by permission of its author, Herbert Breger. The verse-related code is by courtesy of Wayne Sullivan, the author of EDSTANZA \cite{Sul92}, who has kindly supplied more than his original macros.

Since 2011’s Maïeul Rouquette begun to maintain and extend eledmac. As plain TeX is used by little people, and \LaTeX{} by more people eledmac and original EDMAC are more and more distant.

1.1 Overview

The eledmac package, together with \LaTeX{}, provides several important facilities for formatting critical editions of texts in a traditional manner. Major features include:

- automatic stepped line numbering, by page or by section;
- sub-lineation within the main series of line numbers;
- variant readings automatically keyed to line numbers;
- caters for both prose and verse;
• multiple series of the footnotes and endnotes;
• block or columnar formatting of the footnotes;
• simple tabular material may be line numbered;
• indexing keyed to page and line numbers.

eledmac allows the scholar engaged in preparing a critical edition to focus attention wholly on the task of creating the critical text and evaluating the variant readings, text-critical notes and testimonia. \LaTeX\ and eledmac will take care of the formatting and visual correlation of all the disparate types of information.

The original EDMAC can be used as a ‘stand alone’ processor or as part of a process. One example is its use as the formatting engine or ‘back end’ for the output of an automatic manuscript collation program. COLLATE, written by Peter Robinson, runs on the Apple Macintosh, can collate simultaneously up to a hundred manuscripts of any length, and provides facilities for the scholar to tailor the collation interactively. For further details of this and other related work, visit the EDMAC home page at \url{http://www.homepages.ucl.ac.uk/~ucgadku/edmac/index.html}.

EDNOTES \cite{Luc03}, by Uwe Lück and Christian Tapp, is another \LaTeX\ package being developed for critical editions. Unlike eledmac which is based on EDMAC, EDNOTES takes a different (internal) approach and provides a different set of features. For example it provides additional facilities for overlapping lemmas and for handling tables. For more information there is a web site at \url{http://ednotes.sty.de.vu} or email to ednotes.sty@web.de.

The poemscol package \cite{Bur01} by John Burt is designed for typesetting critical editions of collections of poems. I do not know how, or whether, poemscol and eledmac will work together.

Critical authors may find it useful to look at EDMAC, EDNOTES, eledmac, and poemscol to see which best meets their needs.

At the time of writing Peter Wilson knows of two web sites, apart from the EDMAC home page, that have information on eledmac, and other programs.

• Jerónimo Leal pointed me to \url{http://www.guit.sssup.it/latex/critical.html}. This also mentions another package for critical editions called Mauro-\TeX\ \cite{http://www.maurolico.unipi.it/mtex/mtex.htm}. These sites are both in Italian.

• Dirk-Jan Dekker maintains \url{http://www.djdekker.net/ledmac} which is a FAQ for typesetting critical editions and eledmac.

This manual contains a general description of how to use the \LaTeX\ version of EDMAC, namely eledmac (in sections 3 through Appendix A.1); the complete source code for the package, with extensive documentation (in sections 18 and following).
1.2 History

1.2.1 EDMAC

The original version of EDMAC was TEXTED.TEX, written by John Lavagnino in late 1987 and early 1988 for formatting critical editions of English plays.

John passed these macros on to Dominik Wujastyk who, in September–October 1988, added the footnote paragraphing mechanism, margin swapping and other changes to suit his own purposes, making the style more like that traditionally used for classical texts in Latin and Greek (e.g., the Oxford Classical Texts series). He also wrote some extra documentation and sent the files out to several people. This version of the macros was the first to be called EDMAC.

The present version was developed in the summer of 1990, with the intent of adding necessary features, streamlining and documenting the code, and further generalizing it to make it easily adaptable to the needs of editors in different disciplines. John did most of the general reworking and documentation, with the financial assistance of the Division of the Humanities and Social Sciences, California Institute of Technology. Dominik adapted the code to the conventions of Frank Mittelbach’s doc option, and added some documentation, multiple-column footnotes, cross-references, and crop marks.1 A description by John and Dominik of this version of EDMAC was published as ‘An overview of EDMAC: a Plain TeX format for critical editions’, TUGboat 11 (1990), pp. 623–643.

From 1991 through 1994, the macros continued to evolve, and were tested at a number of sites. We are very grateful to all the members of the (now defunct) edmac@mailbase.ac.uk discussion group who helped us with smoothing out bugs and infelicities in the macros. Ron Whitney and our anonymous reviewer at the TUG were both of great help in ironing out last-minute wrinkles, while Ron made some important suggestions which may help to make future versions of EDMAC even more efficient. Wayne Sullivan, in particular, provided several important fixes and contributions, including adapting the Mittelbach/Schöpf ‘New Font Selection Scheme’ for use with PLAIN TeX and EDMAC. Another project Wayne has worked on is a DVI post-processor which works with an EDMAC that has been slightly modified to output \specials. This combination enables you to recover to some extent the text of each line, as ASCII code, facilitating the creation of concordances, an index verborum, etc.

1This version of the macros was used to format the Sanskrit text in volume I of Metarules of Paninian Grammar by Dominik Wujastyk (Groningen: Forsten, 1993).
At the time of writing (1994), we are pleased to be able to say that EDMAC is being used for real-life book production of several interesting editions, such as the Latin texts of Euclid’s *Elements*, an edition of the letters of Nicolaus Copernicus, a Latin translation by Plato of Tivoli of an Arabic astrolabe text, a Latin translation of part II of the Arabic *Algebra* by Abū Kāmil Shujā’ b. Aslam, the Latin *Rithmachia* of Weriner von Tegernsee, a middle-Dutch romance epic on the Crusades, a seventeenth-century Hungarian politico-philosophical tract, an anonymous Latin compilation from Hungary entitled *Sermones Compilati in Studio Generali Quinqueecclesiensi in Regno Ungarie*, the collected letters and papers of Leibniz, Theodosius’s *Spherics*, the German *Algorismus* of Sacrobosco, the Sanskrit text of the *Kāśikāvṛtti* of Vāmana and Jayāditya, and the English texts of Thomas Middleton’s collected works.

1.2.2 eledmac

Version 1.0 of TABMAC was released by Herbert Breger in October 1996. This added the capability for typesetting tabular material.

Version 0.01 of EDSTANZA was released by Wayne Sullivan in June 1992, to help a colleague with typesetting Irish verse.

In March 2003 Peter Wilson started an attempt to port EDMAC from TeX to LaTeX. The starting point was EDMAC version 3.16 as documented on 19 July 1994 (available from CTAN). In August 2003 the TABMAC functions were added; the starting point for these being version 1.0 of October 1996. The EDSTANZA (v0.01) functions were added in February 2004. Sidenotes and regular footnotes in numbered text were added in April 2004.

This port was called ledmac.

Since July 2011, ledmac is maintained by Maïeul Rouquette.

Important changes were put in version 1.0, to make ledmac more easily extensible (see 5.4 p. 25). These changes can trigger small problems with the old

---


3Being prepared at the German Copernicus Research Institute, Munich.

4Being prepared by Menso Folkerts et al., at the Institut für Geschichte der Naturwissenschaften in Munich.

5Richard Lorch, Gerhard Brey et al., at the same Institute.


10Being produced, as was the previous book, by Gyula Mayer in Budapest.

11Leibniz, *Sämtliche Schriften und Briefe*, series I, III, VII, being edited by Dr. H. Breger, Dr. N. Gädeke and others, at the Leibniz-Archiv, Niedersächsische Landesbibliothek, Hannover. (see http://www.nlb-hannover.de/Leibniz)

12Being prepared at Poona and Lausanne Universities.
customization. That is why a new name was selected: *eledmac*. To migrate from ledmac to eledmac, please read Appendix A.2 (p. 258).

### 1.2.3 List of works edited with (e)ledmac

A collaborative list of works edited with (e)ledmac is available on [https://www.zotero.org/groups/critical_editions_typeset_with_edmac_ledmac_and_eledmac/items](https://www.zotero.org/groups/critical_editions_typeset_with_edmac_ledmac_and_eledmac/items). Please add your own edition made with (e)ledmac.

## 2 The eledmac package

*eledmac* is a three-pass package like L\TeX itself. Although your textual apparatus and line numbers will be printed even on the first run, it takes two more passes through L\TeX to be sure that everything gets to its right place. Any changes you make to the input file may similarly require three passes to get everything to the right place, if the changes alter the number of lines or notes. *eledmac* will tell you that you need to make more runs, when it notices, but it does not expend the labor to check this thoroughly. If you have problems with a line or two mismeasured at the top of a page, try running L\TeX once or twice more.

A file may mix numbered and unnumbered text. Numbered text is printed with marginal line numbers and can include footnotes and endnotes that are referenced to those line numbers: this is how you’ll want to print the text that you’re editing. Unnumbered text is not printed with line numbers, and you can’t use eledmac’s note commands with it: this is appropriate for introductions and other material added by the editor around the edited text.

## 3 Options

The package can be loaded with a number of global options which are listed here. It is advised to read the relevant parts of the handbook before reading this section.

- **draft**: underlines lemmas in the main text.
- **ledsecnolinenumber**: is deprecated.
- **nocritical**: disables tools for critical footnotes (\Afootnote, \Bfootnote etc.). If you do not need critical footnotes, this option lets eledmac run faster. It will also preserve room for other packages.
- **noeledsec**: disables tools for \eledsection and related commands ([14.2 p. 49]).
- **noend**: disables tools for end footnotes (\Aendnote, \Bendnote etc.). If you do not need endnotes, this option lets eledmac run faster. It will also preserve room for other packages.
- **nofamiliar**: disables tools for familiar footnotes (\footnoteA, \footnoteB etc.). If you do not need familiar footnotes, this option lets eledmac run faster. It will also preserve room for other packages.
noledgroup  eledmac allows to use of (two or more) critical series of notes and (two or more) new series of normal notes inside minipage and ledgroup environments (see p. 38). However, such features use up computer memory, at the expense of other processing needs. So if you do not need this feature, use noledgroup option. This should make eledmac faster.

nopbinverse  prevents page break inside verses.

noquotation  by default, the quotation environment is redefined inside numbered text. You can disable this redefinition with noquotation (see p. 50).

oldprintnpnumspace  is only to be used if you want to have the (bugged) behavior of \doendnotes of eledmac versions prior to v.1.21.0 (see Appendix A.6.2 p. 260).

parapparatus  by default, the apparatus cannot contain paragraph breaks; this option enables paragraphing inside the apparatus.

series  eledmac defines six levels of notes: A, B, C, D, E, Z. Using all these levels consumes memory space and processing speed. This is why, if your work does not require all of the A-E, Z series, you can narrow down the available number of series. For example, if you only need A and B series, call the package with series={A,B} option.

xindy  and xindy+hyperref are for selecting xindy as the index processor (12.1 p. 44).

widthliketwocolumns  set the width of the text disposed on one column to be the same as the width of the text disposed on two parallel columns with eledpar. This is useful when alternating between normal and parallel typesetting.

4  Text lines and paragraphs numbering

4.1  Text lines numbering

\beginnumbering \endnumbering  Each section of numbered text must be preceded by \beginnumbering and followed by \endnumbering, like:
\beginnumbering (text) \endnumbering

The \beginnumbering macro resets the line number to zero, reads an auxiliary file called \jobname.nn (where \jobname is the name of the main input file for this job, and nn is 1 for the first numbered section, 2 for the second section, and so on), and then creates a new version of this auxiliary file to collect information during this run. The first instance of \beginnumbering also opens a file called \jobname.end to receive the text of the endnotes. \endnumbering closes the \jobname.nn file.
If the line numbering of a text is to be continuous from start to end, then the whole text will be typed between one pair of \texttt{\begin{numbering}} and \texttt{\end{numbering}} commands. But your text will most often contain chapter or other divisions marking sections that should be independently numbered, and these will be appropriate places to begin new numbered sections. \texttt{eledmac} has to read and store in memory a certain amount of information about the entire section when it encounters a \texttt{\begin{numbering}} command, so it speeds up the processing and reduces memory use when a text is divided into a larger number of sections (at the expense of multiplying the number of external files that are generated).

4.2 Paragraphs

4.2.1 Basis

Within a numbered section, each paragraph of numbered text must be marked using the \texttt{\pstart} and \texttt{\pend} commands:

\texttt{\pstart} \texttt{⟨paragraph of text⟩} \texttt{\pend}

Text that appears within a numbered section but isn’t marked with \texttt{\pstart} and \texttt{\pend} will not be numbered.

The following example shows the proper section and paragraph markup, and the kind of output that would typically be generated:

\begin{numbering}
\pstart
This is a sample paragraph, with lines numbered automatically.
\pend

\pstart
This paragraph too has its lines automatically numbered.
\pend

The lines of this paragraph are not numbered.

\pstart
And here the numbering begins again.
\pend
\end{numbering}

4.2.2 Content before specific \pstart and after \pend

Both \texttt{\pstart} and \texttt{\pend} can take a optional argument, in brackets. Its content will be printed before the beginning of \texttt{\pstart} / after the end of \texttt{\pend} instead of the argument of \texttt{\AtEveryPstart} / \texttt{\AtEveryPend}. If you need to start a
**4 Text lines and paragraphs numbering**

\pstart by brackets, or to add brackets after a \pend, just add a \relax between \pstart/\pend and the brackets.

For example, \eledmac does not insert \parskip between paragraphs. This feature allows you to insert it:

\parskip=2baselineskip% Set the skip between paragraphs
\AtEveryPend{vskip\parskip}% Apply after every \Pend

This feature is also useful when typesetting verses (see [p. 34](#)) or \eledpar (see [17.3 p. 53](#)). A \noindent is automatically added before this argument.

### 4.2.3 Content before every \pstart and after every \pend

\AtEveryPstart \AtEveryPend You can use both \AtEveryPstart and \AtEveryPend. Their arguments will be printed before every \pstart begins / after every \pend ends.

### 4.2.4 Producing automatically \pstart...\pend

\autopar You can use \autopar to avoid the nuisance of this paragraph markup and still have every paragraph automatically numbered. The scope of the \autopar command needs to be limited by keeping it within a group, as follows:

\begingroup
  \beginnumbering
  \autopar

  A paragraph of numbered text.
  Another paragraph of numbered text.

  \endnumbering
\endgroup

\autopar fails, however, on paragraphs that start with a \{ or with any other command that starts a new group before it generates any text. Such paragraphs need to be started explicitly, before the new group is opened, using \indent, \noindent, or \leavevmode, or using \pstart itself.\footnote{For a detailed study of the reasons for this restriction, see Barbara Beeton, ‘Initiation rites’, \emph{TUGboat} \textbf{12} (1991), pp. 257–258.}

### 4.2.5 Numbering paragraphs (\pstart)

It is possible to insert a number at every \pstart command. You must use the \numberpstarttrue command to have it. You can stop the numbering with \numberpstartfalse. You can redefine the command \thepstart to
change style. You can change the value of the pstart number by using after \beginnumbering:

\setcounter{numberpstart}{value}

On each \beginnumbering the numbering restarts.

With the \sidepstartnumtrue command, the number of \pstart will be printed inside. In this case, the line number will be not printed.

With the \labelpstarttrue command, a \label added just after a \pstart will refer to the number of this pstart.

4.2.6 Languages written in Right to Left

If you use languages written in right to left, we LuaLaTeX or XeLaTeX, so you must switch text direction \before the \pstart command.

4.2.7 Memory limits

This paragraph is kept for history, but problem described below should not appear with eddmac. eddmac stores a lot of information about line numbers and footnotes in memory as it goes through a numbered section. But at the end of such a section, it empties its memory out, so to speak. If your text has a very long numbered section it is possible that your \LaTeX{} may reach its memory limit. There are two solutions to this. The first is to get a larger \LaTeX{} with increased memory. The second solution is to split your long section into several smaller ones. The trouble with this is that your line numbering will start again at zero with each new section. To avoid this problem, we provide \pausenumbering and \resumenumbering which are just like \endnumbering ... \beginnumbering, except that they arrange for your line numbering to continue across the break. Use \pausenumbering only between numbered paragraphs:

\beginnumbering
\pstart
Paragraph of text.
\pend
\pausenumbering
\resumenumbering
\pstart
Another paragraph.
\pend
\endnumbering

We have defined these commands as two macros, in case you find it necessary to insert text between numbered sections without disturbing the line numbering. But if you are really just using these macros to save memory, you might as well say

\newcommand{\memorybreak}{\pausenumbering\resumenumbering}
and say \textit{memorybreak} between the relevant \texttt{pend} and \texttt{pstart}.

4.3 Lineation commands

4.3.1 Disabling lineation

\texttt{\numberlinefalse}  Line numbering can be disabled with \texttt{\numberlinefalse}. It can be enabled again with \texttt{\numberlinetrue}.

4.3.2 Setting lineation start and step

\texttt{\firstlinenum}  By default, eledmac numbers every 5th line. There are two counters, \texttt{firstlinenum} and \texttt{linenumincrement}, that control this behaviour; they can be changed using \texttt{\firstlinenum{\langle num\rangle}} and \texttt{\linenumincrement{\langle num\rangle}}. \texttt{\firstlinenum} specifies the first line that will have a printed number, and \texttt{\linenumincrement} is the difference between successive numbered lines. For example, to start printing numbers at the first line and to have every other line numbered:

\texttt{\firstlinenum{1} \linenumincrement{2}}

\texttt{\firstsublinenum}  There are similar commands, \texttt{\firstsublinenum{\langle num\rangle}} and \texttt{\sublinenumincrement{\langle num\rangle}} for controlling sub-line numbering. You can define \texttt{\linenumberlist} to specify a non-uniform distribution of printed line numbers. For example:

\texttt{\def\linenumberlist{1,2,3,5,7,11,13,17,19,23,29}}

to have numbers printed on prime-numbered lines only. There must be no spaces within the definition which consists of comma-separated decimal numbers. The numbers can be in any order but it is easier to read if you put them in numerical order. Either omitting the definition of \texttt{\linenumberlist} or following the vacuous definition

\texttt{\def\linenumberlist{}}

the standard numbering sequence is applied. The standard sequence is that specified by the combination of the \texttt{firstlinenum}, \texttt{linenumincrement}, \texttt{firstsublinenum} and \texttt{linenumincrement} counter values.

4.3.3 Setting lineation reset

\texttt{\lineation}  Lines can be numbered either by page, by pstart or by section; you specify this using the \texttt{\lineation{\langle arg\rangle}} macro, where \texttt{\langle arg\rangle} is either \texttt{page}, \texttt{pstart} or \texttt{section}. You may only use this command at places where numbering is not in effect; you can’t change the lineation system within a section. You can change it between sections: they don’t all have to use the same lineation system. The package’s standard setting is \texttt{\lineation{section}}. If the lineation is by pstart, the pstart number will be printed before the line number in the notes.

4.3.4 Setting line number margin

\texttt{\linenummargin}  The command \texttt{\linenummargin{\langle location\rangle}} specifies the margin where the line (or pstart) numbers will be printed. The permissable value for \texttt{\langle location\rangle} is one out of the list \texttt{left}, \texttt{right}, \texttt{inner}, or \texttt{outer}, for example \texttt{\linenummargin{inner}}.
The package’s default setting is \linenummargin{left} to typeset the numbers in the left hand margin. You can change this whenever you’re not in the middle of making a paragraph.

More precisely, the value of \linenummargin used is that in effect at the \pend of a numbered paragraph. Apart from an initial setting for \linenummargin, only change it after a \pend, whereupon it will apply to all following numbered paragraphs, until changed again (changing it between a \pstart and \pend pair will apply the change to all the current paragraph).

4.3.5 Other settings

\leftlinenum \rightlinenum \linenumsep

You can redefine \leftlinenum and \rightlinenum to change the way marginal line numbers are printed in the left and right margins respectively; the initial versions print the number in font \numlabfont (described below) at a distance \linenumsep (initially set to one pica) from the text.

4.4 Changing the line numbers

Normally the line numbering starts at 1 for the first line of a section and steps up by one for each line thereafter. There are various common modifications of this system, however; the commands described here allow you to put such modifications into effect.

\startsub \endsub

You insert the \startsub and \endsub commands in your text to turn sub-lineation on and off. In plays, for example, stage directions are often numbered with sub-line numbers: as line 10.1, 10.2, 10.3, rather than as 11, 12, and 13. Titles and headings are sometimes numbered with sub-line numbers as well.

When sub-lineation is in effect, the line number counter is frozen and the sub-line counter advances instead. If one of these commands appears in the middle of a line, it doesn’t take effect until the next line; in other words, a line is counted as a line or sub-line depending on what it started out as, even if that changes in the middle.

\startlock \endlock

The \startlock command, used in running text, locks the line number at its current value, until you say \endlock. It can tell for itself whether you are in a patch of line or sub-line numbering. One use for line-number locking is in printing poetry: there the line numbers should be those of verse lines rather than of printed lines, even when a verse line requires several printed lines.

\lockdisp

When line-number locking is used, several printed lines may have the same line number, and you have to specify whether you want the number attached to the first printed line or the last, or whether you just want the number printed by them all. (This assumes that, on the basis of the settings of the previous parameters, it is necessary to display a line number for this line.) You specify your preference using \lockdisp{⟨arg⟩}; its argument is a word, either first, last, or all. The package initially sets this as \lockdisp{first}.

\setline \advanceline

In some cases you may want to modify the line numbers that are automatically
calculated: if you are printing only fragments of a work but want to print line numbers appropriate to a complete version, for example. The \setline{(num)} and \advanceline{(num)} commands may be used to change the current line’s number (or the sub-line number, if sub-lineation is currently on). They change both the marginal line numbers and the line numbers passed to the notes. \setline takes one argument, the value to which you want the line number set; it must be 0 or greater. \advanceline takes one argument, an amount that should be added to the current line number; it may be positive or negative.

The \setline and \advanceline macros should only be used within a \pstart...\pend group. The \setlinenum{(num)} command can be used outside such a group, for example between a \pstart and a \pend. It sets the line number to (num). It has no effect if used within a \pstart...\pend group.

Linenumbers are normally printed as arabic numbers. You can use \linenumberstyle{⟨style⟩} to change the numbering style. ⟨style⟩ must be one of:

- Alph Uppercase letters (A… Z).
- alph Lowercase letters (a… z).
- arabic Arabic numerals (1, 2, …)
- Roman Uppercase Roman numerals (I, II, …)
- roman Lowercase Roman numerals (i, ii, …)

Note that with the Alph or alph styles, ‘numbers’ must be between 1 and 26 inclusive.

Similarly \sublinenumberstyle{⟨style⟩} can be used to change the numbering style of sub-line numbers, which is normally arabic numerals.

When inserted into a numbered line the macro \skipnumbering causes the numbering of that particular line to be skipped; that is, the line number is unchanged and no line number will be printed. Note that if you use it in \stanza, you must call it at the beginning of the verse.

When inserted into a numbered line the macro \hidenumbering causes the number for that particular line to be hidden; namely, no line number will print. Note that if you use it in \stanza, you must call it at the beginning of the verse.

5 The apparatus

5.1 Commands

5.1.1 The lemma

\edtext Within numbered paragraphs, all footnotes and endnotes are generated by the \edtext macro:

\edtext{⟨lemma⟩}{⟨commands⟩}
5.1 Commands

The ⟨lemma⟩ argument is the lemma in the main text: \texttt{\edtext} both prints this as part of the text, and makes it available to the ⟨commands⟩ you specify to generate notes.

For example:
\begin{verbatim}
I saw my friend \edtext{Smith}{\Afootnote{Jones C, D.}}
on Tuesday.
\end{verbatim}

The lemma Smith is printed as part of this sentence in the text, and is also made available to the footnote that specifies a variant, Jones C, D. The footnote macro is supplied with the line number at which the lemma appears in the main text.

The ⟨lemma⟩ may contain further \texttt{\edtext} commands. Nesting makes it possible to print an explanatory note on a long passage together with notes on variants for individual words within the passage. For example:
\begin{verbatim}
\edtext{I saw my friend \edtext{Smith}{\Afootnote{Jones C, D.}} on Tuesday.}{\Bfootnote{The date was July 16, 1954.}}
\end{verbatim}

However, \texttt{\edtext} cannot handle overlapping but unnested notes—for example, one note covering lines 10–15, and another covering 12–18; a \texttt{\edtext} that starts in the ⟨lemma⟩ argument of another \texttt{\edtext} must end there, too. (The \texttt{\lemma} and \texttt{linenum} commands may be used to generate overlapping notes if necessary.)

5.1.2 Footnotes

The second argument of the \texttt{\edtext} macro, ⟨commands⟩, may contain a series of subsidiary commands that generate various kinds of notes.

\texttt{\Afootnote} \texttt{\Bfootnote} \texttt{\Cfootnote} \texttt{\Dfootnote} \texttt{\Efootnote} \texttt{\Zfootnote}

Six separate series of the footnotes are maintained; each macro takes one argument like \texttt{\Afootnote{⟨text⟩}}. When all of the six are used, the A notes appear in a layer just below the main text, followed by the rest in turn, down to the Z notes at the bottom. These are the main macros that you will use to construct the critical apparatus of your text.

If you need more series of critical notes, please look at \texttt{5.7.1 p. 34}.

An optional argument can be added before the text of the footnote. Its value is a comma separated list of options. The available options are:

- \texttt{fulllines} to disable \texttt{\twolines} and \texttt{\morethantwolines} features for this note (cf. \texttt{5.4.1 p. 25}).
- \texttt{nonum} to disable line numbering for this note.
- \texttt{nosep} to disable the lemma separator for this note.

Example: \texttt{\Afootnote[nonum]{⟨text⟩}}.
5.1.3 Endnotes

The package also maintains six separate series of endnotes.

If you do not need the endnotes facility, you should use \noend option when loading \eledmac.

The mechanism is similar to the one for footnotes: each macro takes one or more optional arguments and one single argument, like:

\Aendnote[(option)]{(text)}

[<option>] can contain a comma separated list of values. Allowed values are:

- \fulllines to disable \Xendtwolines and \Xendmorethantwolines features for this particular note (cf. 5.4.1 p. 25).
- \nosep to disable the lemma separator for this particular note.

Normally, endnotes are not printed: you must use the \doendnotes{(s)}, where (s) is the letter of the series to be printed. Put this command where you want the corresponding set of endnotes printed.

In this case, all the endnotes of the (s) series are printed, for all numbered section. However, you may want to print the endnotes of one given series covering the first numbered section, then the endnotes of another given series covering the first numbered section, then the endnotes of the first given series covering the second numbered section, then the endnotes of the second given series covering the second numbered section, and so forth. In this case, use \doendnotesbysection{(s)}. For each value of (s), the first call of the command will print the notes for the first series, the second call will print the notes for the second series etc. For example, do:

\section{Endnotes}
\subsection{First text}
\doendnotesbysection{A}
\doendnotesbysection{B}
\subsection{Second text}
\doendnotesbysection{A}
\doendnotesbysection{B}

Note that by default inside endnotes no separator is used between the lemma and the content. However you can use the \Xendlemmaseparator macro to define one (5.4.2 p. 28).

As endnotes may be printed at any point in the document they always start with the page number where they are called. The macro \printnpnum{(num)} is used to print these numbers. Its default definition is:

\newcommand*{\printnpnum}[1]{{p. #1}}

5.1.4 Paragraph in critical apparatus

By default, no paragraph can be made in the notes of critical apparatus. You can allow it by adding the options \parapparatus when loading the package.
If you want to change the lemma that gets passed to the notes, you can do this by using `\lemma{(alternative)}` within the second argument to `\edtext`, before the note commands. The most common use of this command is to abbreviate the lemma that’s printed in the notes. For example:

```
\edtext{I saw my friend
\edtext{Smith}{\Afootnote{Jones C, D.}} on Tuesday.}
{\lemma{I \ldots Tuesday.}
\Bfootnote{The date was July 16, 1954.}}
```

1 I saw my friend
2 Smith on Tuesday.
3 Smith Jones C, D.
4 The date was July 16, 1954.

You can use `\linenum{(arg)}` to change the line numbers passed to the notes. The notes are actually given seven parameters: the page, line, and sub-line number for the start of the lemma; the same three numbers for the end of the lemma; and the font specifier for the lemma. As the argument to `\linenum`, you specify those seven parameters in that order, separated by vertical bars (the `|` character). However, you can retain the value computed by eledmac for any number by simply omitting it; and you can omit a sequence of vertical bars at the end of the argument. For example, `\linenum{|||23}` changes one number, the ending page number of the current lemma.

This command doesn’t change the marginal line numbers in any way; it just changes the numbers passed to the footnotes. Its use comes in situations that `\edtext` has trouble dealing with for whatever reason. If you need notes for overlapping passages that aren’t nested, for instance, you can use `\lemma` and `\linenum` to generate such notes despite the limitations of `\edtext`. If the `<lemma>` argument to `\edtext` is extremely long, you may run out of memory; here again you can specify a note with an abbreviated lemma using `\lemma` and `\linenum`. The numbers used in `\linenum` need not be entered manually; you can use the ‘x-’ symbolic cross-referencing commands below (p. 38) to compute them automatically.

Similarly, being able to manually change the lemma’s font specifier in the notes might be important if you were using multiple scripts or languages. The form of the font specifier is three separate codes separated by `/` characters, giving the family, series, and shape codes as defined within NFSS.

**Changing the names of these commands** The commands for generating the apparatus have been given rather bland names, because editors in different fields have widely divergent notions of what sort of notes are required, where they should be printed, and what they should be called. But this doesn’t mean you have to type `\Afootnote` when you’d rather say something you find more meaningful, like `\variant`. We recommend that you create a series of such aliases and use them instead of the names chosen here; all you have to do is put commands of this form
5.2 Disambiguation of identical words in the apparatus

Sometimes, the same word occurs twice (or more) in the same line. \texttt{eledmac} provides tools to disambiguate references in the critical notes. The lemma will be followed by a reference number if a given word occurs more that once in the same line.

5.2.1 Basic use

To use this tool, you have to mark every occurrence of the potentially ambiguous term with the \texttt{\textbackslash{same}word} command:

\texttt{Lupus \textbackslash{same}word\{aut\} canis \texttt{\textbackslash{edtext}\{\textbackslash{same}word\{aut\}\}\{\textbackslash{Afootnote\{et\}}\} felix}

In this example, \texttt{aut} will be followed, in the critical note, by the exponent 2 if it is printed in the same line as the first \texttt{aut}, but it won’t if it is printed in a different line. The number is printed only after the second run.

5.2.2 Note about input encoding with UTF-8 processor

If you use UTF-8 processor, like \texttt{Xe\TeX} or \texttt{Lua\TeX}, there should not be any glitches. However, pay attention to how characters are encoded. Similar-looking characters may be represented differently in unicode numbering.

For instance, in Greek, \texttt{\textalpha{}} has two possible unicode numbers:

- \texttt{GREEK SMALL LETTER ALPHA (U+03B1) + COMBINING GREEK YPOGEGRAMMENI (U+0345)}
- \texttt{GREEK SMALL LETTER ALPHA WITH YPOGEGRAMMENI (U+1FB3)}

Which unicode number you use depends, many times, on your keyboard configuration (the computer-input system).

Inside \texttt{eledmac}, the \texttt{\textbackslash{same}word} command considers these two unicodes options as different characters. If you use only one unicode number consistently, the distinction will probably make no difference to how your text looks, but \texttt{\textbackslash{same}word} will process the text inaccurately, based on the unicode numbers. To prevent this, do the following:

\footnote{We use \texttt{\newcommand} and \texttt{\newcommandx} instead of classical \texttt{\let} command because the 	exttt{edtable} environments have to modify the notes definition, and we need to use the newest definition of notes. Read the handbook of \texttt{xargs} to know more about \texttt{\newcommandx}.}
5.2 Disambiguation of identical words in the apparatus

- If you use X\raisebox{0.5ex}{e}LaTeX, add this line in your preamble: \texttt{\XeTeXinputnormalization 1}.
- If you use LuaLaTeX, use the \texttt{uninormalize} of Michal Hoftich\textsuperscript{15} with the \texttt{buffer} option set to true.

With these tools, X\raisebox{0.5ex}{e}LaTeX / LuaLaTeX will dynamically normalize unicode input when reading the file. Consequently, you will have no problems with the \texttt{\sameword} command.

5.2.3 Use with \texttt{\lemma} command

If you use the \texttt{\lemma} command, \texttt{eledmac} cannot know to which occurrence of \texttt{\sameword} in the first argument of \texttt{\edtext} a word marked with \texttt{\sameword} in \texttt{\lemma} should refer.

For example in the following example:

```
\begin{verbatim}
some thing
\edtext{\sameword{sw}}
and other \sameword{sw}
and again \sameword{sw}
it is all}\\
\lemma{\sameword{sw} \ldots all}\Afootnote{critical note}.%\end{verbatim}
```

\texttt{eledmac} cannot know if the “sw” in \texttt{\lemma} refers to the word after “thing”, after “other”, or after “again”.

Consequently, you have to tell \texttt{eledmac} which instance of \texttt{\sameword} in the first argument of \texttt{\edtext} you want to reference:

- In the content of \texttt{\lemma}, use \texttt{\sameword} with no optional argument.
- In the first argument of \texttt{\edtext}, use \texttt{\sameword} with the optional argument \texttt{[⟨X⟩]}. \texttt{⟨X⟩} is the depth of the \texttt{\edtext} where the \texttt{\lemma} is used. So if the \texttt{\lemma} is called in a \texttt{\edtext} inside another \texttt{\edtext}, \texttt{⟨X⟩} is equal to 2. If the \texttt{\lemma} is called in a \texttt{\edtext} “of first level”, \texttt{⟨X⟩} is equal to 1. If the lemma is called in both 1 and 2 \texttt{\edtext} depth, \texttt{⟨X⟩} is 1,2. If that word is referenced in the lemma of every \texttt{\edtext} depth, \texttt{⟨X⟩} can also be set to \texttt{\inlemma}.

Note that only words that are actually referenced in a \texttt{\lemma} need the optional argument. Therefore, the first \texttt{\sameword} in the example above should have “1” as its optional argument, to be referenced correctly in the lemma.

Note also that the \texttt{⟨X⟩} does not refer to the level where the \texttt{\sameword} occurs, but to the level of the \texttt{\lemma} that refers to that \texttt{\sameword}. For example:

```
\begin{verbatim}
\edtext{some} \edtext{\sameword[1]{word}}\Afootnote{om. M}
and other \sameword{word}
and again a \sameword{word}
\end{verbatim}
```

\textsuperscript{15}https://github.com/michal-h21/uninormalize
The apparatus

24

The \texttt{sameword} occurs in an \texttt{edtext} of level 2, but since it is referenced by \texttt{lemma} on level 1, it has “1” in the optional argument.

In the following schema, each framed box represents an \texttt{edtext} level. Each number is an occurrence of \texttt{sameword}. After a framed box, the text in superscript represents the content of \texttt{lemma} for that \texttt{edtext} level. The text in subscript at the right of a number represents the content of the optional argument of \texttt{sameword}.

\begin{array}{ccc}
1_{\text{inlemma}} & 2 & 3_2 \text{...} 4_{5_1} \text{...} 5_{1...5} \\
\end{array}

The \texttt{sameword} number 3 is called in a \texttt{lemma} related to an \texttt{edtext} of level 2. It must be marked by “2”.

The \texttt{sameword} number 5 is called in a \texttt{lemma} related to \texttt{edtext} of level 1. It must be marked by “1”.

The \texttt{sameword} number is called in two \texttt{lemmas}: one related to a \texttt{edtext} of level 1, the other related to \texttt{edtext} of level 2. It must be marked by “1,2”. However, as \texttt{lemma} is called only in level 1 and 2, “1,2” could replaced by “inlemma”.

The \texttt{sameword} number “2” is in the first argument of a \texttt{edtext} of level 3, but it has no \texttt{lemma}-command, so there is no need to mark it.

5.2.4 Customizing

You can redefine the \texttt{showwordrank} macro to change the way the number is printed. The default value is

\begin{verbatim}
newcommand{\showwordrank}[2]{% #1\textsuperscript{#2}%
}
\end{verbatim}

5.3 Alternate footnote formatting

If you just launch into \texttt{eledmac} using the commands outlined above, you will get a standard layout for your text and notes. You may be happy to accept this at the very beginning, while you get the hang of things, but the standard layout is not particularly pretty, and you will certainly want to modify it in due course. The package provides ways of changing the fonts and layout of your text, but these are not aimed at being totally comprehensive. They are enough to deal with simple variations from the norm, and to exemplify how you might go on to make more significant changes.

By default, all footnotes are formatted as a series of separate paragraphs in one column. Three other formats are also available for notes, and using these macros you can select a different format for a series of notes.

- \texttt{footparagraph} formats all the footnotes of a series as a single paragraph;
• \foottwocol formats them as separate paragraphs, but in two columns;
• \footthreecol, in three columns.

Each of these macros takes one argument: a letter (between A and E) for the series of notes you want changed. So a text with three layers of notes might begin thus:

\footnormal{A}
\footthreecol{B}
\footparagraph{C}

This would make the A-notes ordinary, B-notes would be in three columns, and the bottom layer of notes would be formed into a paragraph on each page.

5.4 Display options

Since version 1.0, some commands can be used to change the display of the footnotes. All can have an optional argument \[⟨s⟩\], which is the letter of the series — or a list of letters separated by comma — depending on which option is applied.

When a length, noted \[⟨l⟩\], is used, it can be stretchable: a plus b minus c. The final length \[m\] is calculated by \LaTeX{} to have: a + b − c. If you use some relative unity\(^{16}\), it will be relative to fontsize of the footnote, except for commands concerning the place kept by the notes — including blank space.

5.4.1 Control line number printing

By default, the line number is printed in every note. If you want to print it only the first time for a given line number (i.e one time for line 1, one time for line 2 etc.), you can use \numberonlyfirstinline[⟨s⟩].

Use \numberonlyfirstinline[⟨s⟩][false] to disable this ((⟨s⟩ can be empty if you want to disable it for every series).

Suppose you have a lemma on line 2 and a lemma between line 2 and line 3. With \numberonlyfirstinline, the second lemma is considered to be on the same line as the first lemma. But if you use both \numberonlyfirstinline[⟨s⟩] and \numberonlyfirstintwolines[⟨s⟩], the distinction is made. Use \numberonlyfirstintwolines[⟨s⟩][false] to disable this ((⟨s⟩ can be empty if you want to disable it for every series).

If a lemma is printed on two subsequent lines, \eledmac will print the first and the last line numbers. Instead of this, it is also possible to print an abbreviation which stands for “line 1 and subsequent line(s)”. To achieve this, use \twolines[⟨s⟩]{⟨text⟩} and \morethantwolines[⟨s⟩]{⟨text⟩}.

The ⟨text⟩ argument of \twolines will be printed if the lemma is on two lines, and the ⟨text⟩ argument of \morethantwolines will be printed if the lemma is on three or more lines. For example:

\twolines{sq.}  

\footnote{Like \emph{em} which is the width of a mg.}
\textbf{\texttt{\morethantwolines{sqq.}}} \\

Will print “1sq.” for a lemma which falls on lines 1-2 and “1sqq.” for a lemma which falls on lines 1-4.

\texttt{\morethantwolines} \textbf{\texttt{\twolines}} without setting \texttt{\morethantwolines}, the \texttt{(text)} argument of \texttt{\twolines} will be used for lemmas which fall on three or more lines.

However, if you want to use a short form (when the lemma overlaps two lines, but not more than two), use \texttt{\twolinesbutnotmore{(series)\[false\]}}.

It is possible to disable \texttt{\twolinesbutnotmore{(series)\[false\]}} with \texttt{\twolinesbutnotmore{(series)\[false\]}}. When you use lineation by page, the final page number, if different from the initial page number, will not be printed, because the final page number is included in the \texttt{\Xendtwolines} symbol.

\textbf{\texttt{\twolinesonlyinsamepage}} \\

However, you can force print the final page number with \texttt{\twolinesonlyinsamepage{(series)\[false\]}}.

Use \texttt{\twolinesonlyinsamepage{(series)\[false\]}} to disable this.

You can disable \texttt{\twolines} and related for a specific note by using the \texttt{\fulllines} argument in the note macro cf. 5.1.2 p. 19.

\textbf{\texttt{\Endtwolines}} \\

For endnotes, use \texttt{\Endtwolines}; \texttt{\Endmorethantwolines}; \texttt{\Endtwolinesbutnotmore}; \texttt{\Endtwolinesonlyinsamepage}.

\textbf{\texttt{\symlinenum}} \\

For setting a particular symbol in place of the line number, you can use \texttt{\symlinenum{(symbol)}} in combination with \texttt{\numberonlyfirstinline{(series)}}. From the second lemma of the same line, the symbol will be used instead of the line number. Note that any command called in \texttt{(symbol)} must be robust. Use \texttt{\robustify} to robustify a not robust command.

\textbf{\texttt{\nonumberinfootnote{(series)}}} \\

You can use \texttt{\nonumberinfootnote{(series)}} if you don’t want to have the line number in a footnote. To cancel it, use \texttt{\nonumberinfootnote{(series)\[false\]}}.

\textbf{\texttt{\pstartinfootnote{(series)}}} \\

You can use \texttt{\pstartinfootnote{(series)}} if you want to print the pstart number in the footnote, before the line and subline number. Use \texttt{\pstartinfootnote{(series)\[false\]}} to disable this \texttt{(series)} can be empty if you want to disable it for every series). Note that when you change the lineation system, the option is automatically switched:

- If you use lineation by pstart, the option is enabled.
- If you use lineation by section or by page, the option is disabled.

\textbf{\texttt{\pstartinfootnoteteverytime{(series)}}} \\

By default, the pstart number is printed only in the part of text where you have called \texttt{\numberpstarttrue}. We don’t know why you would like to print the pstart number in the notes and not in the main text. However, if you want to do it, you can call \texttt{\pstartinfootnoteteverytime{(series)}}. In this case, the pstart number will be printed every time in footnote.

\textbf{\texttt{\onlypstartinfootnote{(series)}}} \\

In combination with \texttt{\pstartinfootnote{(series)}} you can use \texttt{\onlypstartinfootnote{(series)\[false\]}} if you want to print only the pstart number in the footnote, and not the line and subline number. Use \texttt{\onlypstartinfootnote{(series)\[false\]}} to disable this \texttt{(series)} can be empty if you want to disable it for every series).

\textbf{\texttt{\beforenumberinfootnote{(series)\{l\}}}} \\

With \texttt{\beforenumberinfootnote{(series)\{l\}}}, you can add some space before
5.4 Display options

the line number in a footnote. If the line number is not printed, the space is not either. The default value is 0 pt.
\afternumberinfootnote
With \afternumberinfootnote{⟨s⟩}{⟨l⟩} you can add some space after the line number in a footnote. If the line number is not printed, the space is not either. The default value is 0 pt.
\nonbreakableafternumber
By default, the space defined by \afternumberinfootnote is breakable. With \nonbreakableafternumber{⟨s⟩} it becomes nonbreakable. Use \nonbreakableafternumber{⟨s⟩}[false] to disable this (⟨s⟩ can be empty if you want to disable it for every series).
\beforesymlinenum
With \beforesymlinenum{⟨s⟩}{⟨l⟩} you can add some space before the line symbol in a footnote. The default value is value set by \beforenumberinfootnote.
\aftersymlinenum
With \aftersymlinenum{⟨s⟩}{⟨l⟩} you can add some space after the line symbol in a footnote. The default value is value set by \afternumberinfootnote.
\inplaceofnumber
If no number or symbolic line number is printed, you can add a space, with \inplaceofnumber{⟨s⟩}{⟨l⟩}. The default value is 1 cm.
\boxlinenum
It could be useful to put the line number inside a fixed box: the content of the note will be printed after this box. You can use \boxlinenum{⟨s⟩}{⟨l⟩} to do that. To subsequently disable this feature, use \boxlinenum with length equal to 0 pt. One use of this feature is to print line number in a column, and the note in an other column:
\Xhangindent{1em}
\afternumberinfootnote{0em}
\boxlinenum{1em}
\boxsymlinenum
\boxsymlinenum{⟨s⟩}{⟨l⟩} is the same as \boxlinenum but for the line number symbol.
boxlinenumalign
If you put line number in box, it will be aligned left inside the box. However, you can change it using \boxlinenumalign{⟨s⟩}{⟨text⟩} where ⟨text⟩ can be the following:
L to align left (default value);
R to align right;
C to center.

When using boxlinenum, eledmac put all the line number description in the same box. That is, the same box will contain: the start line number, the dash, and either the end line number or the range symbol (like ff.). However, it is possible to box them in two different boxes.
• \boxstartlinenum{⟨s⟩}{⟨l⟩} will box the start line number in a box of length ⟨l⟩. The content will be put at the right of the box.
• \boxendlinenum{⟨s⟩}{⟨l⟩} will box the dash plus the end line number or the range symbol in a box of length ⟨l⟩. The content will be put at the left of the box.
With these two commands, it is possible to horizontally align the dash of line number when using critical notes, to obtain something like:

1
12-23
24ff.

\boxXendlinenum, \boxXendlinenumalign, \boxXendstartlinenumalign, \boxXendendlinenumalign are the same as, respectively, \boxlinenum and \boxlinenumalign, except in endnotes.

5.4.2 Separator between the lemma and the note

\lemmaseparator For footnotes  By default, in a footnote, the separator between the lemma and the note is a right bracket (\rbracket). You can use \lemmaseparator[⟨s⟩]{⟨lemmaseparator⟩} to change it. The optional argument can be used to specify the series in which it is used. Note that there is a non-breakable space between the lemma and the separator, but a breakable space between the separator and the lemma.

\beforelemmaseparator Using \beforelemmaseparator[⟨s⟩]{⟨l⟩} you can add some space between lemma and separator. If your lemma separator is empty, this space won’t be printed. The default value is 0 em.

\afterlemmaseparator Using \afterlemmaseparator[⟨s⟩]{⟨l⟩} you can add some space between separator and note. If your lemma separator is empty, this space won’t be printed. The default value is 0.5 em.

\nolemmaseparator You can suppress the lemma separator, using \nolemmaseparator[⟨s⟩], which is simply a alias of \lemmaseparator[⟨s⟩]{⟨⟩}.

\inplaceoflemmaseparator With \inplaceoflemmaseparator[⟨s⟩]{⟨l⟩} you can add a space if no lemma separator is printed. The default value is 1 em.

\Xendlemmaseparator For endnotes  By default, there is no separator inside endnotes between the lemma and the content of the note. You can use \lemmaseparator[⟨s⟩]{⟨lemmaseparator⟩} to change this. The optional argument can be used to specify the series in which it is used. An common value of \lemmaseparator is \rbracket. Note that there is a non-breakable space between the lemma and the separator, but a breakable space between the separator and the lemma.

\Xendbeforelemmaseparator Using \Xendbeforelemmaseparator[⟨s⟩]{⟨l⟩} you can add some space between the lemma and the separator. If your lemma separator is empty, this space won’t be printed. The default value is 0 em.

\Xendafterlemmaseparator Using \Xendafterlemmaseparator[⟨s⟩]{⟨l⟩} you can add some space between the separator and the content of the note. If your lemma separator is empty, this space won’t be printed. The default value is 0.5 em.

\Xendinplaceoflemmaseparator With \Xendinplaceoflemmaseparator[⟨s⟩]{⟨l⟩} you can add some space if you chose to remove the lemma separator. The default value is 0.5 em.
5.4 Display options

5.4.3 Font style

\texttt{\textbackslash notenumfont}\{\texttt{(command)}\} is used to change the font style for line numbers in critical footnotes; \texttt{(command)} must be one (or more) switching command, like \texttt{\textbackslash bfseries}.

\texttt{\textbackslash endnotenumfont}\{\texttt{(command)}\} is used to change the font style for line numbers in critical footnotes. \texttt{(command)} must be one (or more) switching command, like \texttt{\textbackslash bfseries}.

\texttt{\textbackslash notenumfontX}\{\texttt{(command)}\} is used to change the font style for note numbers in familiar footnotes. \texttt{(command)} must be one (or more) switching command, like \texttt{\textbackslash bfseries}.

\texttt{\textbackslash notefontsize}\{\texttt{(command)}\} is used to define the font size of critical footnotes of the series. The default value is \texttt{\textbackslash footnotesize}. The \texttt{(command)} must not be a size in pt, but a standard \texttt{\LaTeX} size, like \texttt{\textbackslash small}.

\texttt{\textbackslash notefontsizeX}\{\texttt{(command)}\} is used to define the font size of critical footnotes of the series. The default value is \texttt{\textbackslash footnotesize}. The \texttt{(command)} must not be a size in pt, but a standard \texttt{\LaTeX} size, like \texttt{\textbackslash small}.

\texttt{\textbackslash Xnotefontsize}\{\texttt{(command)}\} is used to define the font size of end critical footnotes of the series. The default value is \texttt{\textbackslash footnotesize}. The \texttt{(command)} must not be a size in pt, but a standard \texttt{\LaTeX} size, like \texttt{\textbackslash small}.

5.4.4 Font of the lemma

\texttt{\textbackslash lemmadisablefontselection}\{\texttt{(s)}\} allows to disable the font of the lemma in note. The \texttt{\textbackslash Xlemmadisablefontselection}[\texttt{(s)}] command allows to disable it for a specific series.

\texttt{\textbackslash endlemmadisablefontselection}\{\texttt{(s)}\} allows to disable the font of the lemma in endnote. The \texttt{\textbackslash Xendlemmadisablefontselection}\{\texttt{(s)}\} command allows to disable it for a specific series.

5.4.5 Styles of notes content

\texttt{\textbackslash parindent}\{\texttt{\textbackslash series}\} to enable indentation.

\texttt{\textbackslash parindentX}\{\texttt{(command)}\} is used to enable indentation.

\texttt{\textbackslash Xhangindent}\{\texttt{(command)}\} which will be applied in the second line of notes. It can help to make distinction between a new note and a break in a note. The default value is 0 pt.

\texttt{\textbackslash hangindent}\{\texttt{(command)}\} which will be applied in the second line of notes. It can help to make a distinction between a new note and a break in a note.
5.4.6 Arbitrary code at the beginning of notes

The three next commands add an arbitrary code at the beginning of notes. As the name’s space is local to the notes, you can use it to redefine some style inside the notes. For example, if you don’t want the pstart number to be in bold, use:

\bhookXnote{\renewcommand{\thepstart}{\arabic{pstart}.}}

\bhookXnote[⟨series⟩]{⟨code⟩} is to be used at the beginning of the critical footnotes.
\bhooknoteX[⟨series⟩]{⟨code⟩} is to be used at the beginning of the familiar footnotes.
\bhookXendnote[⟨series⟩]{⟨code⟩} is to be used at the beginning of the endnotes.

5.4.7 Options for footnotes in columns

Alignment  By default, texts in footnotes in two or three columns are flushed left without hyphenation. However, you can change this with \Xcolalign[⟨s⟩]{⟨code⟩}, for critical footnotes, and \colalignX[⟨s⟩]{⟨code⟩}, for familiar footnotes.

<code> must be one of the following command:

\justifying to have text justified, as usual with \LaTeX. You can also let <code> empty.
\raggedright to have text left aligned, but without hyphenation. That is the default \texttt{eledmac} setting.
\RaggedRight to have text left aligned with hyphenation.
\raggedleft to have text right aligned, but without hyphenation.
\RaggedLeft to have text right aligned with hyphenation.
\centering to have text centered, but without hyphenation.
\Centering to have text centered with hyphenation.

Size of the columns  For the following four macros, be careful that the columns are made from right to left.

\hsizetwocol[⟨s⟩]{⟨l⟩} is used to change width of a column when critical notes are displaying in two columns. Default value is .45 \hsize.
\hsizethreecol[⟨s⟩]{⟨l⟩} is used to change width of a column when critical notes are displaying in three columns. Default value is .3 \hsize.
\hsizetwocolX[⟨s⟩]{⟨l⟩} is used to change width of a column when familiar notes are displaying in two columns. Default value is .45 \hsize.
\hsizethreecolX[⟨s⟩]{⟨l⟩} is used to change width of a column when familiar notes are displaying in three columns. Default value is .3 \hsize.
5.4 Display options

5.4.8 Options for paragraphed footnotes

\afternote You can add some space after a note by using \afternote[⟨s⟩]{⟨l⟩}. The default value is 1em plus 0.4em minus 0.4em.

\parafootsep For paragraphed footnotes (see below), you can choose the separator between each note by using \parafootsep[⟨s⟩]{⟨text⟩}. A common separator is the double pipe (||), which you can set by using \parafootsep{\parallel}. Note that if the symbol defined by \symlinenum must be used at the beginning of a note, the \parafootsep is not used before this note.

\Xragged Text in paragraphed critical notes is justified, but you can use \Xragged[⟨s⟩]+L to make it ragged left, or \Xragged[⟨s⟩]+R to make it ragged right.

\raggedX Text in paragraphed footnotes is justified, but you can use \raggedX[⟨s⟩]+L to make it ragged left, or \raggedX[⟨s⟩]+R to make it ragged right.

5.4.9 Options for block of notes

\txtbeforeXnotes You can add some text before critical notes with \txtbeforeXnotes[⟨s⟩]{⟨text⟩}.

\beforeXnotes You can change the vertical space printed before the rule of the critical notes with \beforeXnotes[⟨s⟩]{⟨l⟩}. The default value is 1.2em plus 0.6em minus 0.6em.

Be careful, the standard \LaTeX footnote rule, which is used by eledmac, decreases by 3pt. This 3pt decrease is not changed by this command.

\beforenotesX You can change the vertical space printed before the rule of the familiar notes with \beforenotesX[⟨s⟩]{⟨l⟩}. The default value is 1.2em plus 0.6em minus 0.6em.

Be careful, the standard \LaTeX footnote rule, which is used by eledmac, decreases 3pt. These 3pt are not changed by this command.

\afterXrule You can change the vertical space printed after the rule of the critical notes with \afterXrule[⟨s⟩]{⟨l⟩}. The default value is 0pt.

Be careful, the standard \LaTeX footnote rule, which is used by eledmac, adds 2.6pt. These 2.6pt are not changed by this command.

\afterruleX You can change the vertical space printed after the rule of the familiar notes with \afterruleX[⟨s⟩]{⟨l⟩}. The default value is 0pt.

Be careful, the standard \LaTeX footnote rule, which is used by eledmac, adds 2.6pt. These 2.6pt are not changed by this command.

\preXnotes You can set the space before the first series of critical notes printed on each page and set a different amount of space for subsequent the series on the page. You can do it with \preXnotes{⟨l⟩}. Default value is 0pt. You can disable this feature by setting the length to 0pt.

\prenotesX You can set the space before the first printed (in a page) series of familiar notes not to be the same as before other series. Default value is 0pt. You can do it with \prenotesX{⟨l⟩}. You can disable this feature by setting the length to 0 pt.

\maxhXnotes By default, one series of critical notes can take 80% of the page size, before being broken to the next page. If you want to change the size use
\maxhnotes\{⟨s⟩\}\{⟨l⟩\}. Be careful: the length can’t be flexible, and is relative to the current font. For example, if you want the note to take, at most, 33 of the text height, do \maxhnotes\{.33\text{textheight}\}.

\maxhnotes\[⟨s⟩\]\{⟨l⟩\} is the same as previous, but for familiar footnotes.

Be careful with the two previous commands. Actually, for technical purposes, one paragraphed note is considered as one block. Consequently, it can’t be broken between two pages, even if you used these commands. The debug is in the todolist.

5.5 Page layout

You should set up the page layout parameters, and in particular the \baselineskip of the footnotes (this is done for you if you use the standard \notefontsetup), before you call any of these macros because their action depends on these; too much or too little space will be allotted for the notes on the page if these macros use the wrong values.\footnote{There is one tiny proviso about using paragraphed notes: you shouldn’t force any explicit line-breaks inside such notes: do not use \par, \break, or \penalty=-10000. If you must have a line-break for some obscure reason, just suggest the break very strongly: \penalty=-9999 will do the trick. 17.5, \textit{p. 136} explains why this restriction is necessary.}

If you use \eledpar \columns macro, you can call:

- \Xnoteswidthliketwocolumns\[⟨s⟩\] to create critical notes with a two-column size width. Use \Xnoteswidthliketwocolumns\[⟨s⟩\]\{false\} to disable it.
- \notesXwidthliketwocolumns\[⟨s⟩\] to create familiar notes with a two-column size width. Use \notesXwidthliketwocolumns\[⟨s⟩\]\{false\} to disable it.

5.5.1 Endnotes in one paragraph

By default, any new endnote starts a new paragraph. Use \Xendparagraph\[⟨series⟩\] to have all end notes of one given series set in one paragraph.

You can add some space after a endnote series by using \Xendafternote\[⟨s⟩\]\{⟨l⟩\}. The default value is 1em plus.4em minus.4em.

you can choose the separator between each note by \Xendsep\[⟨s⟩\]\{⟨text⟩\}. A common separator is the double pipe ($||$), which you can set by using \Xendsep\{$\parallel$\}.

5.6 Fonts

One of the most important features of the appearance of the notes, and indeed of your whole document, will be the fonts used. We will first describe the commands that give you control over the use of fonts in the different structural elements of the document, especially within the notes, and then in subsequent sections specify how these commands are used.
For those who are setting up for a large job, here is a list of the complete set of \eledmac macros relating to fonts that are intended for manipulation by the user: endashchar, fullstop, numlabfont, and rbracket. Line numbers for the main text are usually printed in a smaller font in the margin. The numlabfont macro is provided as a standard name for that font: it is initially defined as
\newcommand{numlabfont}{\normalfont\scriptsize}
You might wish to use a different font if, for example, you preferred to have these line numbers printed using old-style numerals.

A relatively trivial matter relates to punctuation. In your footnotes, there will sometimes be spans of line numbers like this: 12–34, or lines with sub-line numbers like this: 55.6. The en-dash and the full stop are taken from the same font as the numbers, and it all works nicely. But what if you wanted to use old-style numbers, like 12 and 34? These look nice in an edition, but when you use the fonts provided by Plain \TeX they are taken from a math font which does not have the en-dash or full stop in the same places as a text font. If you (or your macros) just typed $\oldstyle 12--34$ or $\oldstyle 55.6$ you would get ‘12–34’ and ‘55▷6’. So we define endashchar and fullstop, which produce an en-dash and a full stop respectively from the normal document font, whatever font you are using for the numbers. These two macros are used in the macros which format the line numbers in the margins and footnotes, instead of explicit punctuation. We also define an rbracket macro for the right square bracket printed at the end of the lemma in many styles of textual notes (including \eledmac’s standard style). For polyglossia, when the lemma is RTL, the bracket automatically switches to a left bracket.

We will briefly discuss select@lemmamfont here because it is important to know about it now, although it is not one of the macros you would expect to change in the course of a simple job. Hence it is ‘protected’ by having the @-sign in its name.

When you use the \text macro to mark a word in your text as a lemma, that word will normally be printed again in your apparatus. If the word in the text happens to be in a font such as italic or bold you would probably expect it to appear in the apparatus in the same font. This becomes an absolute necessity if the font is actually a different script, such as Arabic or Cyrillic. \select@lemmamfont does the work of decoding \eledmac’s data about the fonts used to print the lemma in the main text and calling up those fonts for printing the lemma in the note.

\select@lemmamfont is a macro that takes one long argument—the cluster of line numbers passed to the note commands. This cluster ends with a code indicating what fonts were in use at the start of the lemma. \select@lemmamfont selects the appropriate font for the note using that font specifier.

\eledmac uses \select@lemmamfont in a standard footnote format macro called normalfootfmt. The footnote formats for each of the layers A to E are \let equal to \normalfootfmt. So all the layers of the footnotes are formatted in the same way.
5.7 Changing series

5.7.1 Create a new series

If you need more than six series of critical footnotes you can create extra series, using \texttt{\textbackslash newseries} command. For example to create G and H series \texttt{\textbackslash newseriesG,H}.

5.7.2 Delete series

As the number of series which are defined increases, \texttt{eledmac} gets slower. If you do not need all of the six standard series (A, B, C, D, E, Z), you can load the package with the \texttt{series} option. For example if you need only series A and B, use:

\texttt{\usepackage[series={A,B}]{eledmac}}

5.7.3 Series order

The default series order is the one called with the \texttt{series} option of the package, or, if this option is not used, A, B, C, D, E, Z. Series order determines footnotes order.

However in some specific cases, you need to change the series order at some point inside the document. You can use \texttt{\textbackslash seriesatbegin\{\langle<s>\rangle\}} to pull up a given series \texttt{\langle<s>\rangle} to the beginning, or \texttt{\textbackslash seriesatend\{\langle<s>\rangle\}} to push it down to the end.

6 Verse

In 1992 Wayne Sullivan\textsuperscript{18} wrote the EDSTANZA macros \cite{Sul92} for typesetting verse in a critical edition. More specifically they were for handling poetry stanzas which use indentation to indicate rhyme or metre.

With Wayne Sullivan’s permission the majority of this section has been taken from \cite{Sul92}. Peter has made a few changes to enable his macros to be used in the \LaTeX\ \texttt{eledmac}, and now in \texttt{eledmac} package.

Use \texttt{\stanza} at the start of a stanza. Each line in a stanza is ended by an ampersand (\&), and the stanza itself is ended by putting \& at the end of the last line.

Lines within a stanza may be indented. The indents are integer multiples of the length \texttt{\stanzaindentbase}, whose default value is 20pt.

In order to use the stanza macros, \textbf{one must set the indentation values.} First the value of \texttt{\stanzaindentbase} should be set, unless the default value 20pt is desired. Every stanza line indentation is a multiple of this.

To specify these multiples one invokes, for example \texttt{\setstanzaindents\{3,1,2,1,2\}}.

\textsuperscript{18}Department of Mathematics, University College, Dublin 4, Ireland
The numerical entries must be whole numbers, 0 or greater, separated by commas without embedded spaces. The first entry gives the hanging indentation to be used if the stanza line requires more than one print line.

If it is known that each stanza line will fit on more than one print line, then this first entry should be 0; \TeX does less work in this case, but no harm ensues if the hanging indentation is not 0 but is never used.

If you want the hanging verse to be flush right, you can use \texttt{\textbackslash hanginsymbol}:

see p. 6.4 p. 36.

Enumeration is by stanza lines, not by print lines. In the above example the lines are indented one unit, two units, one unit, two units, with 3 units of hanging indentation in case a stanza line is too long to fit on one print line.

\section{6.1 Repeating stanza indents}

Since version 0.13, if the indentation is repeated every \textit{n} verses of the stanza, you can define only the \textit{n} first indentations, and say they are repeated, defining the value of the \texttt{stanzaindentsrepetition} counter at \textit{n}. For example:

\begin{verbatim}
\setstanzaindents{5,1,0}
\setcounter{stanzaindentsrepetition}{2}
\end{verbatim}

is like

\begin{verbatim}
\setstanzaindents{0,1,0,1,0,1,0,1,0,1,0}
\end{verbatim}

Be careful: the feature is changed in eledmac 1.5.1. See Appendix A.3 p. 259.

If you don’t use the \texttt{stanzaindentsrepetition} counter, make sure you have at least one more numerical entry in \texttt{\setstanzavalues} than the number of lines in the stanza.

If you want to disable this feature again, just put the counter to 0:

\begin{verbatim}
\setcounter{stanzaindentsrepetition}{0}
\end{verbatim}

The macros make no restriction on the number of lines in a stanza. Stanza indentation values (and penalty values) obey \TeX’s grouping conventions, so if one stanza among several has a different structure, its indentations (penalties) may be set within a group; the prior values will be restored when the group ends.

\section{6.2 Manual stanza indent}

You can set the indent of some specific verse by calling \texttt{\stanzaindent\{\textit{value}\}} at the beginning of the verse, before any other character. In this case, the indent defined by \texttt{\setstanzaindents} for this verse is skipped, and \texttt{\{\textit{value}\}} is used instead.
If you use the mechanism of indent repetition, the next verse will be printed as it should be even if the current verse would have its normal indent value. In other words, using \stanzaindent in a verse does not shift the indent repetition.

However, if you want to shift the indent repetition, so the next verse has the indent normally used for the current verse, use \stanzaindent* instead of \stanzaindent.

### 6.3 Stanza breaking

When the stanzas run over several pages, it is often desirable that page breaks should arise between certain lines in the stanza, so a facility for including penalties after stanza lines is provided. If you are satisfied with the page breaks, you need not set the penalty values.

The command
\setstanzapenalties{1,5000,10100,5000,0}
results in a penalty of 5000 being placed after the first and third lines of the stanza, and a penalty of −100 after the second.

The first entry “1” is a control value. If it is zero, then no penalties are passed on to \TeX, which is the default. Values between 0 and 10000 are penalty values; values between 10001 and 20000 have 10000 subtracted and the result is given as a negative penalty. The mechanism used for indentations and penalties requires unsigned values less than 32768. No penalty is placed after the last line, so the final ,0 in then example above could be omitted. The control sequence \endstanzaeextra can be defined to include a penalty. A penalty of 10000 will prevent a page break; such a penalty is included automatically where there is stanza hanging indentation. A penalty of −10000 (corresponding to the entry value 20000 in this context) forces a page break. Values in between act as suggestions as to the desirability of a page break at a given line. There is a subtle interaction between penalties and glue, so it may take some adjustment of skips and penalties to achieve the best results.

### 6.4 Hanging symbol

It's possible to insert a symbol in each line of hanging verse, as in French typography for ‘[’. To insert in cledmac, redefine macro \hangingsymbol with this code:

\renewcommand{\hangingsymbol}{\[,}

You can also use it to force hanging verse to be flush right:

\renewcommand{\hangingsymbol}{\protect\hfill}
6.5 Long verse and page break

If you want to prevent page breaks inside long verses, use the option \nopbinverse when loading package, or use \lednopbinversetrue. Read \[6] p. \[50] for further details.

6.6 Various tools

\ampersand If you need to print an & symbol in a stanza, use the \ampersand macro, not \& which will end the stanza.

\endstanzaextra The macro \endstanzaextra, if it is defined, is called at the end of a stanza. You could define this, for example, to add extra space between stanzas (by default there is no extra space between stanzas); if you are using the memoir class, it provides a length \stanzaskip which may come in handy.

\startstanzahook Similarly, if \startstanzahook is defined, it is called by \stanza at the start. This can be defined to do something.

\flagstanza Putting \flagstanza\[\langle\text\rangle\] at the start of a line in a stanza (or elsewhere) will typeset \langle\text\rangle at a distance \langle\text\rangle before the line. The default \langle\text\rangle is \stanzaindentbase.

For example, to put a verse number before the first line of a stanza you could proceed along the lines:

\newcounter{stanzanum}
\setcounter{stanzanum}{0}
\newcommand*{\startstanzahook}{\refstepcounter{stanzanum}}
\newcommand{\numberit}{\flagstanza{\thestanzanum}}
...
\stanza
\numberit First line...&
   rest of stanza\&

\stanza
\numberit First line, second stanza...

6.7 Hanging symbol

It’s possible to insert a symbol on each line of hanging verse, as in French typography for ‘\[’. To insert in eledmac, redefine macro \hangingsymbol with this code:

\renewcommand{\hangingsymbol}{\[,\]}

6.8 Text before/after verses

It is possible to add text, like a subtitle, before or after verse:

- \stanza command can take a optional argument (in brackets). Its content will be printed before the stanza.
• \& can be replaced by \newverse with two optional arguments (in brackets). The first will be printed after the current verse, the second before the next verse.

• \& can take a optional argument (in brackets). Its content will be printed after the stanza.

7 Grouping

In a \texttt{minipage} environment \LaTeX{} changes \texttt{footnote} numbering from arabic to alphabetic and puts the footnotes at the end of the minipage. You can put numbered text with critical footnotes in a minipage and the footnotes are set at the end of the minipage.

You can also put familiar footnotes (see section \ref{section:footnote}) in a minipage but unlike with \texttt{footnote} the numbering scheme is unaltered.

Minipages, of course, aren’t broken across pages. Footnotes in a \texttt{ledgroup} environment are typeset at the end of the environment, as with minipages, but the environment includes normal page breaks. The environment makes no change to the textwidth so it appears as normal text; it just might be that footnotes appear in the middle of a page, with text above and below.

The \texttt{ledgroupsized} environment is similar to \texttt{ledgroup} except that you must specify a width for the environment, as with a minipage.

\begin{verbatim}
\begin{ledgroupsized}{\textwidth}
\end{ledgroupsized}
\end{verbatim}

The \texttt{ledgroupsized} environment is effectively the same as \texttt{\begin{ledgroup}}

8 Crop marks

The \texttt{eledmac} package does not provide crop marks. These are available with either the \texttt{memoir} class \cite{Wil02} or the \texttt{crop} package.

9 Cross referencing

The package provides a simple cross-referencing facility that allows you to mark places in the text with labels, and generate page and line number references to those places elsewhere using those labels.
9.1 Basic use

\edlabel First you place a label in the text using the command \edlabel{(lab)}. (lab) can be almost anything you like, including letters, numbers, punctuation, or a combination—anything but spaces; you might say \edlabel{toves-3}, for example.\footnote{More precisely, you should stick to characters in the \TeX categories of ‘letter’ and ‘other’.}

\edpageref Elsewhere in the text, either before or after the \edlabel, you can refer to its location via \edpageref{(lab)}, or \edlineref{(lab)}\footnote{Previously, the \edlineref command was \lineref. But some packages also define \lineref. That is why you should use \edlineref instead of \lineref. \eledmac defines \lineref as equal to \edlineref, except if one package has also defined a \lineref command.}, \sublineref{(lab)}, or \pstartref{(lab)}. These commands will produce, respectively, the page, line, sub-line and pstart on which the \edlabel{(lab)} command occurred.

An \edlabel command may appear in the main text, or in the first argument of \edtext, but not in the apparatus itself. But \edpageref, \edlineref, \sublineref, \pstartref commands can also be used in the apparatus to refer to \edlabels in the text.

The \edlabel command works by writing macros to \TeX aux file. You will need to process your document through \TeX twice in order for the references to be resolved.

You will be warned if you say \edlabel{foo} and foo has been used as a label before. The ref commands will return references to the last place in the file marked with this label. You will also be warned if a reference is made to an undefined label. (This will also happen the first time you process a document after adding a new \edlabel command: the auxiliary file will not have been updated yet.)

If you want to refer to a word inside an \edtext{...}{...} command, the \edlabel should be defined inside the first argument, e.g.,

The \edtext{creature}{\edlabel{elephant} was quite unafraid}{\Afootnote{Of the mouse, that is.}}

If you add the \edlabel inside some \Xfootnote command, it will refer to that note, and a suffix n will be added to the reference. You can redefine this suffix by redefining the command \ledinnotemark. Its actual definition is:

\newcommand{\ledinnotemark}[1]{#1\textit{n}}

Where \#1 stands for the reference. However, there are situations in which you’ll want \eledmac to return a number without displaying any warning messages about undefined labels or the like: if you want to use the reference in a context where \TeX is looking for a number, such a warning will lead to a complaint that the number is missing. This is the case for references used within the argument to \linenum, for example. For this situation, three variants of the reference commands, with the x prefix, are supplied: \xpageref, \xlineref, \xsblineref and \xpstartref. They have these limitations:
• They will not tell you if the label is undefined.

• They must be preceded in the file by at least one of the four other cross-reference commands—e.g., a \edtext{foo} command, even if you never refer to that label—since those commands can all do the necessary processing of the .aux file, and the \x... ones cannot.

• When hyperref is loaded, the hyperref link won’t be added. (Indeed, it’s not a limitation, but a feature.)

The macros \xxref and \edmakelabel let you manipulate numbers and labels in ways which you may find helpful in tricky situations.

The \xxref{⟨lab1⟩}{⟨lab2⟩} command generates a reference to a sequence of lines, for use in the second argument of \edtext. It takes two arguments, both of which are labels: e.g., \xxref{mouse}{elephant}. It calls \linenum (q.v., 5.1.4 p. [21] above) and sets the beginning page, line, and sub-line numbers to those of the place where \edlabel{mouse} was placed, and the ending numbers to those where \edlabel{elephant} occurs.

Sometimes the \edlabel command cannot be used to specify exactly the page and line desired—for example, if you want to refer to a page and line number in another volume of your edition. In such cases, you can use the \edmakelabel{⟨lab⟩}{⟨numbers⟩} macro so that you can ‘roll your own’ label. For example, if you say \edmakelabel{elephant}{10|25|0} you will create a new label, and a later call to \edpageref{elephant} would print ‘10’ and \lineref{elephant} would print ‘25’. The sub-line number here is zero. It is usually best to collect your \edmakelabel statements near the top of your document, so that you can see them at a glance.

9.2 Normal LaTeX cross-referencing

The normal \label, \ref and \pageref macros may be used within numbered text, and operate in the familiar fashion.

9.3 References to lines commented in the apparatus

You may want to make a cross-reference to a passage that is referred to by \edtext. eledmac provides specific tools for this scenario.

If you use \applabel{⟨label⟩} inside the second argument of a \edtext, eledmac will add a \edlabel at the beginning and end of the marked passage. The label at the beginning of the passage will have the title \texttt{⟨label⟩}:start, while the label at the end will have the title \texttt{⟨label⟩}:end.

If you use \linenum{5.1.4 p. [21]} to refer to these labels, eledmac will use your line settings to refer to the passage.

You can also use \appref{⟨label⟩} and \apprefwithpage{⟨label⟩} to refer to these lines. The first one will print the lines as they are printed in the critical footnotes, while the second will print the lines as they are printed in endnotes.

If you redefine \apprefprefixsingle, its content will be printed before the
line numbers of a \appref-reference. If you redefine \apprefprefixmore, its content will be printed before the line numbers, if you refer to more than one line.

For example, you may use:

\renewcommand{\apprefprefixsingle}{line~}
\renewcommand{\apprefprefixmore}{lines~}

Note that if \apprefprefixmore is empty, \apprefprefixsingle will be used in any case.

If you use \twolines, \morethantwolines, \twolinesbutnotmore and/or \twolinesonlyinsamepage \cite{5.4.1 p. 25} without the optional series argument, the setting will also be available for \appref.

The commands \twolinesappref{⟨text⟩}, \morethantwolinesappref{⟨text⟩}, \twolinesbutnotmoreappref \twolinesonlyinsamepageappref can also be used, if you only want to change the reference style of \appref.

It is possible to disable this setting for a specific \appref command by using \appref{fulllines}{⟨label⟩}.

The commands \twolinesapprefwithpage{⟨text⟩}, \morethantwolinesapprefwithpage{⟨text⟩}, \twolinesbutnotmoreappref \twolinesonlyinsamepageapprefwithpage can also be used, if you only want to change the reference style of \apprefwithpage.

It is possible to disable this setting for a specific \apprefwithpage command by using \apprefwithpage{fulllines}{⟨label⟩}.

10 Side notes

The \marginpar command does not work in numbered text. Instead the package provides for non-floating sidenotes in either margin.

\ledinnernote{⟨text⟩} will put ⟨text⟩ into the inner margin level with where the command was issued. Similarly, \ledouternote{⟨text⟩} puts ⟨text⟩ in the outer margin.

\ledleftnote\ledrightnote \ledsidenote\sidemargin\sidemarkorgan
\ledlsnotewidth\ledrsnotewidth

\ledinnernote\ledouternote
\ledleftnote\ledrightnote
\ledsidenote\sidemargin
\sidemarkorgan
\ledlsnotewidth\ledrsnotewidth

The left sidenote text is put into a box of width \ledlsnotewidth and the
right text into a box of width \ledrsnotewidth. These are initially set to the value of \marginparwidth.

By default, Sidenotes are placed to align with the last line of the note to which it refers. If you want them to be placed to align with the first line of the note to which it refers, use \leftnoteupfalse (for left note) and/or \rightnoteupfalse (for right note).

The texts are put a distance \ledlsnotesep (or \ledrsnotesep) into the left (or right) margin. These lengths are initially set to the value of \linenumsep.

These macros specify how the sidenote texts are to be typeset. The initial definitions are:

\newcommand*{\ledlsnotefontsetup}{\raggedleft\footnotesize}% left
\newcommand*{\ledrsnotefontsetup}{\raggedright\footnotesize}% right

These can of course be changed to suit.

If you have two or more sidenotes for the same line, they are separated by a comma. But if you want to change this separator, you can redefine the macro \sidenotesep.

11 Familiar footnotes

The footmisc package by Robin Fairbairn has an option whereby sequential footnote marks in the text can be separated by commas like so. As a convenience \eledmac provides this automatically.

\multfootsep is used as the separator between footnote markers. Its default definition is:
\providecommand*{\multfootsep}{\textsuperscript{,}} and can be changed if necessary.

As well as the standard \LaTeX\footnote macros, the package also provides six series of additional footnotes called \footnoteA through \footnoteZ. These have the familiar marker in the text, and the marked text at the foot of the page can be formatted using any of the styles described for the critical footnotes. Note that the ‘regular’ footnotes have the series letter at the end of the macro name whereas the critical footnotes have the series letter at the start of the name.

\footnormalX Each of the \foot...X macros takes one argument which is the series letter (e.g., B). \footnormalX is the typical footnote format. With \footparagraphX the series is typeset as one paragraph, with \foottwocolX the notes are set in two columns, and are set in three columns with \footthreecolX.
\thefootnoteA As well as using the \foot...X macros to specify the general footnote arrangement for a series, each series uses a set of macros for styling the marks. The mark numbering scheme is defined by the \thefootnoteA macro; the default is:
\renewcommand*{\thefootnoteA}{\arabic{footnoteA}}

The appearance of the mark in the text is controlled by \bodyfootmarkA which is defined as:
\newcommand*{\bodyfootmarkA}{% 
  \hbox{\textsuperscript{\normalfont\@nameuse{@thefnmarkA}}}\}

The command \footfootmarkA controls the appearance of the mark at the start
of the footnote text. It is defined as:
\newcommand*{\footfootmarkA}{\textsuperscript{\@nameuse{@thefnmarkA}}} 

There are similar command triples for the other series.

Additional footnote series can be easily defined: you just have to use
\newseries, defined above (see \textit{5.7.1} p. 34).

11.1 Position of the familiar footnotes

There is a historical incoherence in \texttt{eledmac}. The familiar footnotes are before
the critical footnotes in a normal page, but after in a minipage or in a ledgroup.
However, it is possible to change the relative position of both types of footnotes.
If you want to have familiar footnotes after critical footnotes in a normal page,
use:
\fnpos{critical-familiar}

Or, if you want a minipage or ledgroup to have critical footnotes after familiar
footnotes, use:
\mpfnpos{familiar-critical}

12 Indexing

\index \LaTeX provides the \texttt{\index{⟨item⟩}} command for specifying that ⟨item⟩ and
the current page number should be added to the raw index (idx) file. The
\texttt{\edindex{⟨item⟩}} macro can be used in numbered text to specify that ⟨item⟩
and the current page & linenumber should be added to the raw index file.

Note that the file .idx will contain the right reference only after the third run,
because of the internal indexing mechanism of \texttt{eledmac}. That means you must
first run three times (Xe/Lua)\LaTeX, then run makeindex and finally run again
(Xe/Lua)\LaTeX to get an index with the right page numbers.

If the \texttt{memoir} class or the \texttt{imakeidx} or \texttt{indextools} package is used then the
macro takes an optional argument, which is the name of a raw index file. For
example \texttt{\edindex[line]{item}} will use \texttt{line.idx} as the raw file instead of
\texttt{\jobname.idx}.

The minimal version of \texttt{imakeidx} package to be used is the version 1.3a uploaded
on CTAN on 2013/07/11.

Be careful with the order of package loading and index declaration. You must
use this order:
1. Load imakeidx or indextools .
2. Load eledmac.
3. Declare the index with the macro \makeindex of imakeidx/indextools.

\pagelinesep  The page & linenumber combination is written as page\pagelinesep line, where the default definition is \newcommand{\pagelinesep}{-} so that an item on page 3, line 5 will be noted as being at 3--5. You can renew \pagelinesep to get a different separator.

- is the default separator used by the MAKEINDEX program.

Consequently, if you want to use an other \pagelinesep, you have to configure your .ist index style file. For example if you use : as separator\footnote{For further detail, you can read \url{http://tex.stackexchange.com/a/32783/7712}}.

\begin{verbatim}
page_compositor ";"
delim_r ":";
\end{verbatim}

\edindexlab Read the MAKEINDEX program’s handbook about the .ist file. The \edindex process uses a \label/\ref mechanism to get the correct line number. It automatically generates labels of the form \label{\edindexlab N}, where N is a number, and the default definition of \edindexlab is:

\newcommand*{\edindexlab}{$&}

in the hopes that this will not be used by any other labels (\edindex’s labels are like \label{$&27}). You can change \edindexlab to something else if you need to.

\section*{12.1 Using xindy}

Should you decide to use \texttt{xindy} instead of \texttt{makeindex} to transform your .idx files into .ind files, you must use some specific configuration file (.xdy) so that \texttt{xindy} can understand \texttt{eledmac} reference syntax of which the scheme is:

\begin{verbatim}
pagenumber-linenumber
\end{verbatim}

An example of such a file is provided in the "examples" folder. Read the xindy handbook to learn how to use it.\footnote{Or, for people who read French, read \url{http://geekographie.maieul.net/174}}

This file also provides, with an explanation, the settings that are needed to put eledmac lines numbers in parenthesis, in order to make a better distinction between line numbers and page ranges.

In any case, you must load \texttt{eledmac} with the \texttt{xindy} option, in order to generate a .xdy file which is specific to your document. This file is needed by the .xdy example file which is in the "examples" folder. Its default name is eledmac-markup-attr.xdy, but you can change it by using your own as an argument of the \texttt{xindy+hyperref} option.

If you chose to use both \texttt{xindy} and the \texttt{hyperref} package, you must do three more things:

1. Use \texttt{xindy+hyperref} option when loading the \texttt{eledmac} package. When you run (Xe/Lua)\TeX with this option, a .xdy configuration file will be generated with all the settings needed to allow internal hyperlinking in each index entry which is created by \texttt{edindex}.
2. Use `hyperindex=false` option when loading hyperref.

3. Uncomment — by removing the semicolons at the beginning of the relevant lines — some lines in the `<code>.xdy</code>` file provided in the “examples” folder in order to restore internal links in the index to be used by the standard `index` command.

13 Tabular material

LaTeX’s normal `tabular` and `array` environments cannot be used where line numbering is being done; more precisely, they can be used but with odd results, so don’t use them. However, cledmac provides some simple tabulation environments that can be line numbered. The environments can also be used in normal unnumbered text.

There are six environments; the `edarray*` environments are for math and `edarrayl` `edarrayc` `edarrayr` `edtabularl` `edtabularc` `edtabularr` for text entries. The final `l`, `c`, or `r` in the environment names indicate that the entries will be flushleft (`l`), centered (`c`) or flushright (`r`). There is no means of specifying different formats for each column, nor for specifying a fixed width for a column. The environments are centered with respect to the surrounding text.

```latex
\begin{edtabularc}
  1 & 2 & 3 \\
  a & bb & ccc \\
  AAA & BB & C
\end{edtabularc}
```

Entries in the environments are the same as for the normal `array` and `tabular` environments but there must be no ending `\` at the end of the last row. There must be the same number of column designators (the `\&`) in each row. There is no equivalent to any line drawing commands (such as `\hline`). However, unlike the normal environments, the `ed...` environments can cross page breaks.

Macros like `\edtext` can be used as part of an entry. For example:

```latex
\begin{numbering}
\pstart
\begin{edtabularl}
\textbf{\Large I} wish I was a little bug\index{bug} &
\textbf{\Large I} eat my peas with honey\index{honey} \&
& With whiskers \edtext{round}{\texttt{Around}} my tummy &
& I've done it all my life. \&
& I'd climb into a honey\index{honey} pot &
& It makes the peas taste funny \&
& And get my tummy gummy.\index{gummy} &
& But it keeps them on the knife.
\end{edtabularl}
\end{numbering}
```

23These are the recommended lines to provide the best possible compatibility between hyperref and xindy, even without using `eledmac`. 
produces the following parallel pair of verses.

1 I wish I was a little bug
2 With whiskers round my tummy I've done it all my life.
3 I'd climb into a honey pot It makes the peas taste funny
4 And get my tummy gummy. But it keeps them on the knife.

The distance between the columns is controlled by the length \texttt{\edtabcolsep}. \texttt{\spreadmath{(math)} typesets \texttt{(math)} but the \texttt{(math)} has no effect on the calculation of column widths. \texttt{\spreadtext{text}} is the analogous command for use in \texttt{edtabular} environments.

\begin{edarrayl}
1 & 2 & 3 & 4 \\
& \spreadmath{F+G+C} & & \\
an & bb & ccc & dddd \\
\end{edarrayl}

\texttt{\edrowfill{\langle\text{start}\rangle}{\langle\text{end}\rangle}{\langle\text{fill}\rangle}} fills columns number \texttt{\langle\text{start}\rangle} to \texttt{\langle\text{end}\rangle} inclusive with \texttt{\langle\text{fill}\rangle}. The \texttt{\langle\text{fill}\rangle} argument can be any horizontal ‘fill’. For example \texttt{\hrulefill} or \texttt{\upbracefill}.

Note that every row must have the same number of columns, even if some would not appear to be necessary.

The \texttt{\edrowfill} macro can be used in both tabular and array environments. The typeset appearance of the following code is shown below.
You can also define your own ‘fill’. For example:

\newcommand*{\upbracketfill}{% \
vrule height 4pt depth 0pt\hrulefill\vrule height 4pt depth 0pt}\}

is a fill like \upbracefill except it has the appearance of a (horizontal) bracket instead of a brace. It can be used like this:

\begin{edarrayc}
1 & 2 & 3 & 4 \\
a & 2 & 3 & 4 & d \\
A & B & C & D
\end{edarrayc}

\edatleft\[
\left<symbol>\right.\]
\edatright\[
\left<symbol>\right.\]
\edbeforetab\{\}
\edaftertab\{\}

\begin{edarrayc}
& 1 & 2 & 3 & \\
& 4 & 5 & 6 & \\
\edbeforetab{Before}\{\}(B) & 1 & 3 & 6 & \\
\edaftertab{After}\{\}(B) & 1 & 5 & 0
\end{edarrayc}

\begin{edarrayl}
A \ & 1 & 2 & 3 \\
\edbeforetab{Before}\{\}(B) & 1 & 3 & 6 \\
C \ & 1 & 4 \ & \edaftertab{After}\{\}(B) \ & D \ & 1 & 5 & 0
\end{edarrayl}
The macro \texttt{\textbackslash edvertline\{\textit{height}\}} draws a vertical line \textit{\texttt{\textbackslash edvertline}} high (contrast this with \texttt{\textbackslash edatright} where the size argument is half the desired height).

\begin{verbatim}
\begin{edarrayr}
a & b & C & d & \\
v & w & x & y & \\
m & n & o & p & \\
k & L & cvb & \textbackslash edvertline\{4pc\}
\end{edarrayr}
\end{verbatim}

\texttt{\textbackslash edvertdots} macro is similar to \texttt{\textbackslash edvertline} except that it produces a vertical dotted instead of a solid line.

14 Sectioning commands

14.1 Sectioning commands without line numbers or critical notes

The standard sectioning commands (\texttt{\textbackslash chapter}, \texttt{\textbackslash section} etc.) can be used inside numbered text. In this case, you must call them as an optional argument of \texttt{\textbackslash pstart} \texttt{\textbackslash pend} \texttt{\textbackslash pstart\{section\}}:

\begin{verbatim}
\pstart\{section\}
 Pstart content.
\pend
\end{verbatim}

The line which contains them won’t be numbered, and you can’t add critical notes inside.

14.2 Sectioning commands with line numbering and critical notes

In the past (between versions 1.1.0 and 1.12.0), these following commands were provided:
14.2 Sectioning commands with line numbering and critical notes

- \ledchapter[⟨text⟩]{⟨critical text⟩}
- \ledchapter*
- \ledsection[⟨text⟩]{⟨critical text⟩}
- \ledsection*
- \ledsubsection[⟨text⟩]{⟨critical text⟩}
- \ledsubsection*
- \ledsubsubsection[⟨text⟩]{⟨critical text⟩}
- \ledsubsubsection*

These commands are deprecated, and won’t be maintained anymore, because of a bad concept. Since version 1.12.0, you have to use the following commands:

- \eledchapter[⟨text⟩]{⟨critical text⟩}
- \eledchapter*
- \eledsection[⟨text⟩]{⟨critical text⟩}
- \eledsection*
- \eledsubsection[⟨text⟩]{⟨critical text⟩}
- \eledsubsection*
- \eledsubsubsection[⟨text⟩]{⟨critical text⟩}
- \eledsubsubsection*

Which are equivalent to the \LaTeX commands. Each individual command must be called alone in a \pstart...\pend:

\begin{verbatim}
\pstart
\eledsection*[xxxx\ledsidenote{section}]
\pend
\pstart
\eledsubsection*[xxxx\ledsidenote{sub}]
\pend
\pstart
normal text
\pend
\end{verbatim}

At the first run, you will see only the text. It’s normal. At the second run, you will see the formatting. And consequently, at the third run, you will see the table of contents.

For technical reasons, the page break before \elechapter can’t be added automatically. You have to insert it manually via \beforeeledchapter, which must be called outside of a numbered section. If you aren’t going to have any \eledxxx commands, then load \eledmac with \noeledsec option. That will suppress the generation of unneeded \eledsec file, keep memory and make \eledmac faster.
15 Quotation environments

The quotation and quote environment can be used so that same definition/note appears both inside and outside a numbered section. The typographical consequences will resemble the outside numbered sections, based on the styles of the book class. However, if you use a package that redefines these environments, these redefinitions won’t be available inside the numbered section. You must open any quotation environments inside a \start\pend block, not outside. A quotation environment MUST not be opened immediately after a \pstart and MUST not be closed immediately before a \pend.

In some cases, you don’t want these environments to be redefined in numbered sections. You can load the package with the option noquotation to prevent this redefinition.

16 Page breaks

Eledmac and eledpar break pages automatically. However, you may sometimes want to either force page breaks or prevent them. The packages provide two macros:

- \ledpb adds a page break.
- \lednopb prevents a page break, by adding one line to the current page if needed.

These commands have effect only at the second run.

These two commands take effect at the beginning of line in which they are called. For example, if you call \ledpb at l. 444, the l. 443 will be at the p. n, and the l. 444 at the p. n + 1. However you can change the behavior, and decide they will have effect after the end of the line, adding \ledpsetting{after} at the beginning of your file (better: in your preamble). With the previous example, the l. 444 will be at the p. n and the l. 445 will be at the p. n + 1.

If you are using eledpar to typeset parallel pages you must use \lednopb on both sides in the two corresponding lines. This is especially important when you are using stanzas; otherwise the pages will run out of sync. You can also decide to prevent page breaks between two lines of a long verse. To do this, use nopinverse when loading package, or add \lednopbinversetrue in the beginning of your file (better: in your preamble). This feature works only with verse of 2 lines, not more. It works at the third run, or at fourth run with eledpar. By default, when a long verse runs normally between two pages, a page break will be placed at the beginning of the verse. However, if you have added \ledpsetting{after}, the page break will be placed at the end of the long verse, and the page containing the long verse will have one extra line.
17 Miscellaneous

\extensionchars When the package assembles the name of the auxiliary file for a section, it prefixes \extensionchars to the section number. This is initially defined to be empty, but you can add some characters to help distinguish these files if you like; what you use is likely to be system-dependent. If, for example, you said \renewcommand{\extensionchars}{!}, then you would get temporary files called \jobname.1, \jobname.2, etc.

\ifledfinal The package can take options. The option ‘final’, which is the default is for final typesetting; this sets \ifledfinal to TRUE. The other option, ‘draft’, may be useful during earlier stages and sets \ifledfinal to FALSE.

\showlemma The lemma within the text is printed via \showlemma{lemma}. Normally, or with the ‘final’ option, the definition of \showlemma is:
\newcommand*{\showlemma}[1]{#1}
so it just produces its argument. With the ‘draft’ option it is defined as
\newcommand*{\showlemma}[1]{\textit{#1}}
so that its argument is typeset in an italic font, which may make it easier to check that all lemmas have been treated.

If you would prefer some other style, you could put something like this in the preamble:
\ifledfinal\else
  \renewcommand{\showlemma}[1]{\textbf{#1}}% or simply ...[1]{#1}
\fi

17.1 Known and suspected limitations

In general, eledmac’s system for adding marginal line numbers breaks anything that makes direct use of the \TeX insert system, which includes marginpars, footnotes and floats.

However, you can use both \footnote and the familiar footnote series notes in numbered text. A \marginpar in numbered text will throw away its contents and send a warning message to the terminal and log file, but will do no harm.

\parshape cannot be used within numbered text, except in a very restricted way.

\ballast \TeX is a three-pass system, but even after a document has been processed three times, there are some tricky situations in which the page breaks decided by \TeX never settle down. At each successive run, eledmac may oscillate between two different sets of page decisions. To stop this happening, should it arise, Wayne Sullivan suggested the inclusion of the quantity \ballast. The amount of \ballast will be subtracted from the penalties which apply to the page breaks calculated on the previous run through \TeX, thus reinforcing these breaks. So if you find your page breaks oscillating, say
\setcounter{ballast}{100}
or some such figure, and with any luck the page breaks will settle down. Luckily, this problem doesn’t crop up at all often.
The restriction on explicit line-breaking in paragraphed footnotes, mentioned in a footnote on p. 17, and described in more detail on p. 25.5, really is a nuisance if that is something you need to do. There are some possible solutions, described by Michael Downes, but this area remains unsatisfactory.

\TeX{} has a reputation for putting things in the wrong margin after a page break. The \texttt{eledmac} package does nothing to improve the situation — in fact it just makes it more obvious if numbered text crosses a page (or column) boundary and the numbers are meant to flip from side to side. Try and keep the numbers in the same margin all the time. Another aspect of \TeX{}’s page breaking mechanism is that when numbering lines by the page, the first few numbers after a page break may continue as though the lines were still on the previous page.

If you can’t resist flipping the numbers or numbering by the page, then you might find that judicious use of \texttt{\pageparbreak} may help if numbering goes awry across a page (or column) break. It tries to force \TeX{} into partitioning the current paragraph into two invisibly joined paragraphs with a page break between them. Insert the command between the last word on one page and the first word on the next page. If later you change something earlier in the document the natural page break may be in a different place, and you will have to adjust the location of \texttt{\pageparbreak} accordingly.

For paragraphed footnotes \TeX{} has to estimate the amount of space required. If it underestimates this then the notes may get too long and run off the bottom of the text block. \texttt{\footfudgefiddle} can be increased from its default 64 (say to 68) to increase the estimate. You have to use \texttt{\renewcommand} for this, like:

\begin{verbatim}
\renewcommand{\footfudgefiddle}{68}
\end{verbatim}

Help, suggestions and corrections will be gratefully received.

\section*{17.2 Use with other packages}

Because of \texttt{eledmac}’s complexity it may not play well with other packages. In particular \texttt{eledmac} is sensitive to commands in the arguments to the \texttt{\edtext} and \texttt{\*footnote} macros (this is discussed in more detail in section 22, and in particular the discussion about \texttt{\noexpand} and \texttt{\morenoexpand}). You will have to see what works or doesn’t work in your particular case.

It is possible that \texttt{eledmac} and the \texttt{hyperref} package may work together. I have not tried this combination but past experience with \texttt{hyperref} suggests that cooperation is unlikely; \texttt{hyperref} changes many \TeX{} internals and \texttt{eledmac} does things that are not normally seen in \TeX{}.

If you want to use the option \texttt{bottom} of the \texttt{footmisc} package, you must load this package \texttt{before} the \texttt{eledmac} package.

\texttt{\morenoexpand} You can define the macro \texttt{\morenoexpand} to modify macros that you call within \texttt{\edtext}. Because of the way \texttt{eledmac} numbers the lines the arguments to \texttt{\edtext} can be processed more than once and in some cases a macro should only be processed once. One example is the \texttt{\colorbox} macro from the \texttt{color} package, which you might use like this:

\begin{verbatim}
\ldots \edtext{\colorbox{mycolor}{lemma}}{\Afootnote{\ldots\colorbox{\ldots}}}
\end{verbatim}
If you actually try this\textsuperscript{24} you will find \LaTeX whinging ‘\texttt{Missing { inserted’}, and then things start to fall apart. The trick in this case is to specify either:

\begin{verbatim}
\newcommand{\morenoexpands}{\let\colorbox=0}
\end{verbatim}

or

\begin{verbatim}
\makeatletter
\newcommand{\morenoexpands}{\let\colorbox@secondoftwo}
\makeatother
\end{verbatim}

(\@secondoftwo is an internal \LaTeX macro that takes two arguments and throws away the first one.) The first incantation lets color show in both the main text and footnotes whereas the second one shows color in the main text but kills it in the lemma and footnotes. On the other hand if you use \texttt{textcolor} instead, like

... \edtext{\textcolor{mycolor}{lemma}}{\Afootnote{...\textcolor{mycolor}...}}

there is no need to fiddle with \texttt{morenoexpands} as the color will naturally be displayed in both the text and footnotes. To kill the color in the lemma and footnotes, though, you can do:

\begin{verbatim}
\makeatletter
\newcommand{\morenoexpands}{\let\textcolor@secondoftwo}
\makeatother
\end{verbatim}

It took me a little while to discover all this. If you run into this sort of problem you may have to spend some time experimenting before hitting on a solution.

17.3 Parallel typesetting

Peter Wilson has developed the \texttt{ledpar} package as an extension to \texttt{eledmac} specifically for parallel typesetting of critical texts. This also cooperates with the \texttt{babel} / \texttt{polyglossia} packages for typesetting in multiple languages. The package has been called \texttt{eledpar} since September 2012.

He also developed the \texttt{ledarab} package for handling parallel Arabic text in critical editions. However, this package is not maintained by Maïeul Rouquette. You should use the capabilities of a modern \TeX processor, like \texttt{Xe(La)TeX}

\textsuperscript{24}Reported by Dirk-Jan Dekker in the CTT thread ‘Incompatibility of “color” package’ on 2003/08/28.
18 Implementation overview

We present the `eledmac` code in roughly the order in which it’s used during a run of TeX. The order is *exactly* that in which it’s read when you load the `eledmac` package, because the same file is used to generate this manual and to generate the LaTeX package file. Most of what follows consists of macro definitions, but there are some commands that are executed immediately—especially at the start of the code. The documentation generally describes the code from the point of view of what happens when the macros are executed, though. As each macro is introduced, its name is printed in the margin.

We begin with the commands you use to start and stop line numbering in a section of text (Section 19). Next comes the machinery for writing and reading the auxiliary file for each section that helps us count lines, and for creating list macros encoding the information from that file (Section 21); this auxiliary file will be read at the start of each section, to create those list macros, and a new version of the file will be started to collect information from the body of the section.

Next are commands for marking sections of the text for footnotes (Section 22), followed by the macros that take each paragraph apart, attach the line numbers and insertions, and send the result to the vertical list (Section 23). The footnote commands (Section 25) and output routine (Section 36) finish the main part of the processing; cross-referencing (Section 37) and endnotes (Section 32) complete the story.

In what follows, macros with an `@` in their name are more internal to the workings of `eledmac` than those made up just of ordinary letters, just as in Plain TeX (see *The TeXbook*, p. 344). You are meant to be able to make free with ordinary macros, but the `@` ones should be treated with more respect, and changed only if you are pretty sure of what you are doing.

19 Preliminaries

We try and use `@d` in macro names to help avoid name clashes, but this is not a hard and fast rule. For example, if an original EDMAC macro includes `edmac` we will simply change that to `eledmac`.

Announce the name and version of the package, which is targeted for LaTeX2e.

```latex
1 \langle+code\rangle
2 \NeedsTeXFormat{LaTeX2e}
3 \ProvidesPackage{eledmac}[2015/05/29 v1.23.2 LaTeX port of EDMAC]

Generally, these are the modifications to the original. EDMAC code:

- Replace as many `\def`'s by `\newcommand`'s as possible to avoid overwriting \LaTeX macros.
- Replace user-level \TeX counts by \LaTeX counters.
- Use the \LaTeX font handling mechanisms.
- Use \LaTeX messaging and file facilities.
```
19.1 Package options

Use this to remember which option is used, set and execute the options with final as the default.

\parapparatusfalse
\RequirePackage{xkeyval}
\DeclareOptionX{series}[A,B,C,D,E,Z]{\xdef\defaultseries{#1}}
\DeclareOptionX{noeledsec}{\@noeledsectrue}
\DeclareOptionX{nocritical}{\nocrITICAL@true}
\DeclareOptionX{nofamiliar}{\nofamiliar@true}
\DeclareOptionX{noledgroup}{\noledgroup@true}
\DeclareOptionX{noend}{\let\l@dend@open@gobble\let\l@d@end\relax\let\l@dend@close\relax\global\let\l@dend@stuff=\relax\global\chardef\l@d@end=16\noend@true}
\DeclareOptionX{noquotation}{\noquotation@true}
\DeclareOptionX{oldprintnpnumspace}{\oldprintnpnumspace@true}
\DeclareOptionX{final}{\ledfinaltrue}
\DeclareOptionX{draft}{\ledfinalfalse}
\DeclareOptionX{parapparatus}{\parapparatus@true}
\DeclareOptionX{nopbinverse}{\lednopbinversetrue}
\DeclareOptionX{ledsecnolinenumber}{\ledsecnolinenumbertrue}
\DeclareOptionX{widthliketwocolumns}{\widthliketwocolumnstrue}
\DeclareOptionX{xindy}{eledmac-markup-attr.xdy}{\AtBeginDocument{\immediate\openout\eledmac@xindy@out=#1}\newwrite\eledmac@xindy@out\xindy@true\gdef\eledmacmarkuplocrefdepth{\depth 1}\AtEndDocument{\immediate\closeout\eledmac@xindy@out}}
\DeclareOptionX{xindy+hyperref}{
Use the starred form of \texttt{ProcessOptions} which executes options in the order listed in the source file: class options, then listed package options, so a package option can override a class option with the same name. This was suggested by Dan Luecking in the \texttt{ctt} thread \textit{Class/package option processing}, on 27 February 2004.

\vspace{1em}

\section{Loading packages}

Loading package \texttt{xargs} to declare commands with optional arguments. \texttt{Etoolbox} is also used to make code clearer - for example, in dynamic command names (which can replace \texttt{csmart} etc.). Use \texttt{suffix} to declare commands with a starred version, \texttt{xstring} to work with strings, \texttt{ifluatex} and \texttt{ifxetex} to test if \LaTeX{} or \XeLaTeX{} is running, and \texttt{ragged2e} to manage ragged for paragraphed notes.

\vspace{1em}

\section{Boolean flags}

\begin{verbatim}
define a flag for if the memoir class has been used.\texttt{\newif\ifl@dmemoir}  
\@ifclassloaded{memoir}{\l@dmemoirtrue}{\l@dmemoirfalse}  

\texttt{\ifl@imakeidx} Define a flag for if the imakeidx package has been used.\newif\ifl@imakeidx  
\@ifpackageloaded{imakeidx}{\l@imakeidxtrue}{\l@imakeidxfalse}  

\texttt{\ifl@indextools} Define a flag for if the indextools package has been used.\newif\ifl@indextools  
\@ifpackageloaded{indextools}{\l@indextoolstrue}{\l@indextoolsfalse}  
\end{verbatim}
\if@RTL The \if@RTL is defined by the bidi package, which is sometimes loaded by \textit{polyglossia}. But we define it as well if the bidi package is not loaded.
\ifdef{\if@RTL}{}{\newif\if@RTL}
\if@RTL The \if@RTL is defined by the bidi package, which is sometimes loaded by \textit{polyglossia}. But we define it if the bidi package is not loaded.
\ifdef{\if@RTL}{}{\newif\if@RTL}

19.4 Messages

All the messages are grouped here as macros. This saves TeX’s memory when the same message is repeated and also lets them be edited easily.

\eledmac@warning Write a warning message.
\newcommand{\eledmac@warning}[1]{\PackageWarning{eledmac}{#1}}

\eledmac@error Write an error message.
\newcommand{\eledmac@error}[2]{\PackageError{eledmac}{#1}{#2}}
\led@err@NumberingStarted
\led@err@NumberingNotStarted
\led@err@NumberingShouldHaveStarted
\newcommand{\led@err@NumberingStarted}{\eledmac@error{Numbering has already been started}{\@ehc}}
\newcommand{\led@err@NumberingNotStarted}{\eledmac@error{Numbering was not started}{\@ehc}}
\newcommand{\led@err@NumberingShouldHaveStarted}{\eledmac@error{Numbering should already have been started}{\@ehc}}
\led@err@edtextoutsidepstart
\newcommand{\led@err@edtextoutsidepstart}{\eledmac@error{\string\edtext\space outside numbered paragraph (\pstart\ldots\pend)}{\@ehc}}
\led@mess@NotesChanged
\newcommand{\led@mess@NotesChanged}{\typeout{eledmac reminder: }\typeout{ The number of the footnotes in this section has changed since the last run.}\typeout{ You will need to run LaTeX two more times before the footnote placement}\typeout{ and line numbering in this section are correct.}}
\led@mess@SectionContinued
\newcommand{\led@mess@SectionContinued}[1]{\message{Section #1 (continuing the previous section)}}
\led@err@LineationInNumbered
\newcommand{\led@err@LineationInNumbered}{\eledmac@error{You can’t use \texttt{\string\lineation} space within a numbered section}{\@ehc}}
\newcommand*{\led@warn@BadLineation}{%}
\newcommand*{\led@warn@BadLinenummargin}{%}
\newcommand*{\led@warn@BadLockdisp}{%}
\newcommand*{\led@warn@BadSublockdisp}{%}
\newcommand*{\led@warn@NoLineFile}{%}
\newcommand*{\led@warn@LineFileObsolete}{%}
\newcommand*{\led@warn@BadAdvancelineSubline}{%}
\newcommand*{\led@warn@BadAdvancelineLine}{%}
\newcommand*{\led@warn@BadSetline}{%}
\newcommand*{\led@warn@BadSetlinenum}{%}
\newcommand*{\led@err@PstartNotNumbered}{%}
\newcommand*{\led@err@PstartInPstart}{%}
\newcommand*{\led@err@PendNotNumbered}{%}
\newcommand*{\led@err@AutoparNotNumbered}{%}
\newcommand*{\led@err@NumberingWithoutPstart}{%}
\newcommand*{\led@err@NumberingWithoutPstart}{%}
\newcommand*{\led@macro@warning}{Bad \string\lineation\space argument}}
\newcommand*{\led@macro@warning}{Bad \string\linenummargin\space argument}}
\newcommand*{\led@macro@warning}{Bad \string\lockdisp\space argument}}
\newcommand*{\led@macro@warning}{Bad \string\sublockdisp\space argument}}
\newcommand*{\led@macro@warning}{Can't find line-list file #1}}
\newcommand*{\led@macro@warning}{Line-list file #1 was obsolete. We have not read it. Please run LaTeX again}}
\newcommand*{\led@macro@warning}{Bad \string\setline\space argument}}
\newcommand*{\led@macro@warning}{Bad \string\setlinenum\space argument}}
\newcommand*{\led@macro@error}{\string\pstart\space must be used within a
numbered section}{\@ehc}}
\newcommand*{\led@macro@error}{\string\pstart\space encountered while another
string\pstart\space was in effect}{\@ehc}}
\newcommand*{\led@macro@error}{\string\pend\space must be used within a
numbered section}{\@ehc}}
\newcommand*{\led@macro@error}{\string\pend\space must follow a \string\pstart}{\@ehc}}
\newcommand*{\led@macro@error}{\string\autopar\space must be used within a
numbered section}{\@ehc}}
\newcommand*{\led@macro@error}{\string\beginnumbering...\string\endnumbering\space without \string\pstart}
19.4 Messages

\led@warn@BadAction

140 \newcommand*{\led@warn@BadAction}{% 
141 \eledmac@warning{Bad action code, value \next@action.}}

\led@warn@DuplicateLabel
\led@warn@AppLabelOutEdtext
\led@warn@RefUndefined

142 \newcommand*{\led@warn@DuplicateLabel}{% 
143 \eledmac@warning{Duplicate definition of label `#1' on page \the\pageno.}}
144 \newcommand*{\led@warn@AppLabelOutEdtext}{% 
145 \eledmac@warning{\string\applabel\space outside of \string\edtext\space `#1' on page \the\pageno.}}
146 \newcommand*{\led@warn@RefUndefined}{% 
147 \eledmac@warning{Reference `#1' on page \the\pageno\space undefined.} 
148 \Using `000'.}}

\led@warn@NoMarginpars

149 \newcommand*{\led@warn@NoMarginpars}{% 
150 \eledmac@warning{You can't use \string\marginpar\space in numbered text}}

\led@warn@BadSidenotemargin

151 \newcommand*{\led@warn@BadSidenotemargin}{% 
152 \eledmac@warning{Bad \string\sidenotemargin\space argument}}

\led@warn@NoIndexFile

153 \newcommand*{\led@warn@NoIndexFile}{% 
154 \eledmac@warning{Undefined index file #1}}

\led@warn@AddfootinsXobsolete
\led@warn@AddfootinsObsolete

155 \newcommand{\led@warn@AddfootinsXObsolete}{% 
156 \eledmac@warning{AddfootinsX is obsolete in eledmac 1.0. Use newseries instead.}}
157 \end@warn@AddfootinsXObsolete
158 \newcommand{\led@warn@AddfootinsObsolete}{% 
159 \eledmac@warning{Addfootins is obsolete in eledmac 1.0. Use newseries instead.}}
160 \end@warn@AddfootinsObsolete

\led@warn@SeriesStillExist

161 \newcommand{\led@warn@SeriesStillExist}{% 
162 \eledmac@warning{Series #1 is still existing !}}
163 \end@warn@SeriesStillExist

\led@err@ManySidenotes
\led@err@ManyLeftnotes
\led@err@ManyRightnotes

164 \newcommand{\led@err@ManySidenotes}{% 
165 \ifdefRco10% 
166 \eledmac@warning{\itemcount@\space sidenotes on line \the\line@numR\space p. \the\page@numR} 
167 \else% 
168 \eledmac@warning{\itemcount@\space sidenotes on line \the\line@num\space p. \the\page@num}\% 
169 \fi% 
170 \} 
171 \newcommand{\led@err@ManyLeftnotes}{% 
172 \ifdefRco10%
Preliminaries
Here, we define some commands which gobble their arguments.

19.6 Miscellaneous commands

\showlemma \showlemma{⟨lemma⟩} typesets the lemma text in the body. It depends on the option.
\if@prevvalue\newcommand*{\showlemma}{#1}\else\newcommand*{\showlemma}{\underline{#1}}\fi

\linenumberlist The code for the \linenumberlist mechanism was given to Peter Wilson by Wayne Sullivan on 2004/02/11.
Initialize it as \empty
\let\linenumberlist=\empty

In imitation of \LaTeX, we create a couple of scratch counters.
\section@num You use \begin{numbering} and \end{numbering} to begin and end a line-numbered section of the text; the pair of commands may be used as many times as you
like within one document to start and end multiple, separately line-numbered sections. \LaTeX will maintain and display a `section number' as a count named \texttt{section@num} that counts how many \texttt{beginnumbering} and \texttt{resumenumbering} commands have appeared; it needn’t be related to the logical divisions of your text.

\begin{verbatim}
\section
\end{verbatim}

Each section will read and write an associated `line-list file', containing information used to do the numbering; the file will be called \texttt{(jobname).nn}, where \texttt{nn} is the section number. However, you may direct that an extra string be added before the \texttt{nn} in that filename, in order to distinguish these temporary files from others: that string is called \texttt{extensionchars}. Initially it’s empty, since different operating systems have greatly varying ideas about what characters are permitted in file names. So \texttt{\renewcommand{\extensionchars}{-}} gives temporary files called \texttt{jobname.-1, jobname.-2}, etc.

\begin{verbatim}
\newcount
\section@num
\section@num=0
\let\extensionchars=\empty
\ifnumbering
\numberingtrue
\numberingfalse
\beginnumbering
\initnumbering@reg
\beginnumbering
\initnumbering@reg
\end{verbatim}

In preparation for the \texttt{eledpar} package, these are related to the `left' text of parallel texts (when \texttt{\ifl@dpairing} is TRUE). They are explained in the \texttt{eledpar} manual.

\begin{verbatim}
\ifnumberingR
\ifl@dpairing
\ifl@dpaging
\ifl@dprintingpages
\ifl@dprintingcolumns
\ifl@dprintingcolumnsfalse
\l@dpairingtrue
\ifl@dprintingcolumnsfalse
\l@dpairingtrue
\ifl@dprintingcolumnsfalse
\l@dpairingtrue
\beginnumbering
\initnumbering@reg
\beginnumbering
\initnumbering@reg
\end{verbatim}

\texttt{\beginnumbering} begins a section of numbered text. When it’s executed we increment the section number, initialize our counters, send a message to your terminal, and call macros to start the lineation machinery and endnote files. The initializations here are trickier than they look. \texttt{\line@list@stuff} will use all of the counters that are zeroed here when it assembles the line-list and other
lists of information about the lineation. But it will do all of this locally and within
a group, and when it’s done the lists will remain but the counters will return to
zero. Those same counters will then be used as we process the text of this section,
but the assignments will be made globally. These initializations actually apply
to both uses, though in all other respects there should be no direct interaction
between the use of these counters and variables in the two processing steps.
For parallel processing:

• zero \@dnumpstartsL — the number of chunks to be processed.

• set \ifpstart\text{L} to FALSE.

\newcommand*{\beginnumbering}{\%
  \ifnumbering
    \led@err@NumberingStarted
  \fi
  \global\numberingtrue
  \global\advance\section@num 1
  \initnumbering@reg
  \message{Section \the\section@num }%
  \line@list@stuff{\jobname.\extensionchars\the\section@num}\
  \l@dend@stuff
  \setcounter{pstart}{1}
  \ifl@dpairing
    \global\@dnumpstartsL \z@
  \else
    \begingroup
      \initnumbering@sectcmd
      \ifwidthliketwocolumns%
        \csuse{setwidthliketwocolumns@columns@position}\
        \csuse{setpositionliketwocolumns@columns@position}\
      \fi%
  \fi
  \gdef\eled@sections@@{}%
  \if@noeled@sec
    \else
      \InputIfFileExists{\jobname.eledsec\the\section@num}{}{}\makeatother%
      \immediate\openout\eled@sectioning@out=\jobname.eledsec\the\section@num\relax%
      \fi%
  \\fi
\}

The tools for section’s title commands are called:

• Define old (deprecated) sectioning commands.

• Define an empty list of pstart number where sectioning commands are called.

• Input auxiliary file with the description of section titles.

• Open the same auxiliary file to write in.
\endnumbering must follow the last text for a numbered section. It takes care of notifying you when changes have been noted in the input that require running the file through again to move everything to the right place.

\endnumbering
\pausenumbering The \pausenumbering macro is just the same as \endnumbering, but with the \ifnumbering flag set to true, to show that numbering continues across the gap.\(^\text{25}\)

\resumenumbering The \resumenumbering macro is a bit more involved, but not much. It does most of the same things as \beginnumbering, but without resetting the various counters. Note that no check is made by \resumenumbering to ensure that \pausenumbering was actually invoked.

21 Line counting

21.1 Choosing the system of lineation

Sometimes you want line numbers that start at 1 at the top of each page; sometimes you want line numbers that start at 1 at each \pstart; other times you want line numbers that start at 1 at the start of each section and increase regardless of page breaks. \eledmac can do it either way, and you can switch from one to the other.

\(^\text{25}\)Our thanks to Wayne Sullivan, who suggested the idea behind these macros.
within one work. But you have to choose one or the other for all line numbers and line references within each section. Here we will define internal codes for these systems and the macros you use to select them.

\ifbypstart@ The \ifbypage@ and \ifbypstart@ flag specify the current lineation system:

• line-of-page: \bypage@ = false and \bypage@ = true.

• line-of-pstart: \bypage@ = true and \bypage@ = false.

eledmac will use the line-of-section system unless instructed otherwise.

350 \newif\ifbypage@
351 \newif\ifbypstart@

\lineation \lineation{⟨word⟩} is the macro you use to select the lineation system. Its argument is a string: either page or section or pstart.

352 \newcommand*{\lineation}{[1]{{% 353 \ifnumbering
354 \led@err@LineationInNumbered
355 \else
356 \def\@tempa{#1}\def\@tempb{page}% 357 \ife\@tempa\@tempb%
358 \global\bypage@true
359 \global\bypstart@false
360 \unless\ifnocritical% 361 \pstartinfootnote[][false]% 362 \fi% 363 \else
364 \def\@tempb{pstart}% 365 \ife\@tempa\@tempb%
366 \global\bypage@false
367 \global\bypstart@true
368 \unless\ifnocritical% 369 \pstartinfootnote% 370 \fi%
371 \else
372 \def\@tempb{section}% 373 \ife\@tempa\@tempb%
374 \global\bypage@false
375 \global\bypstart@false
376 \unless\ifnocritical% 377 \pstartinfootnote[][false]% 378 \fi%
379 \else
380 \led@warn@BadLineation
381 \fi
382 \fi
383 \fi}}
You call \texttt{\linenummargin\{word\}} to specify which margin you want your line numbers in; it takes one argument, a string. You can put the line numbers in the same margin on every page using \texttt{left} or \texttt{right}; or you can use \texttt{inner} or \texttt{outer} to get them in the inner or outer margins. (These last two options assume that even-numbered pages will be on the left-hand side of every opening in your book.) You can change this within a numbered section, but the change may not take effect just when you'd like; if it's done between paragraphs nothing surprising should happen.

The selection is recorded in the count \texttt{\line@margin}: 0 for left, 1 for right, 2 for outer, and 3 for inner.

The following counters tell \texttt{eledmac} which lines should be printed with line numbers. \texttt{firstlinenum} is the number of the first line in each section that gets a number; \texttt{linenumincrement} is the difference between successive numbered lines. The initial values of these counters produce labels on lines 5, 10, 15, etc. \texttt{linenumincrement} must be at least 1.

\begin{verbatim}
\newcounter{firstlinenum}
\setcounter{firstlinenum}{5}
\end{verbatim}
\newcounter{linenumincrement}
\setcounter{linenumincrement}{5}
\c@firstsublinenum
\c@sublinenumincrement

The following parameters are just like firstlinenum and linenumincrement, but for sub-line numbers. sublinenumincrement must be at least 1.
\newcounter{firstsublinenum}
\setcounter{firstsublinenum}{5}
\newcounter{sublinenumincrement}
\setcounter{sublinenumincrement}{5}
\firstlinenum
\linenumincrement
\firstsublinenum
\sublinenumincrement

These macros can be used to set the corresponding counters.
\newcommand*{\firstlinenum}[1]{\setcounter{firstlinenum}{#1}}
\newcommand*{\linenumincrement}[1]{\setcounter{linenumincrement}{#1}}
\newcommand*{\firstsublinenum}[1]{\setcounter{firstsublinenum}{#1}}
\newcommand*{\sublinenumincrement}[1]{\setcounter{sublinenumincrement}{#1}}
\lockdisp
\lock@disp
\l@dgetlock@disp

When line locking is being used, the \lockdisp{⟨word⟩} macro specifies whether a line number—if one is due to appear—should be printed on the first printed line or on the last, or by all of them. Its argument is a word, either first, last, or all. Initially, it is set to first.
\lock@disp encodes the selection: 0 for first, 1 for last, 2 for all.
\newcount{\lock@disp}
\newcommand{\lockdisp}[1]{\ifnum#1=1 \global\lock@disp=1 \else \global\lock@disp=0 \fi}
\newcommand{\l@dgetlock@disp}[1]{\edef\@tempa{#1}\edef\@tempb{first}\ifx\@tempa\@tempb \global\lock@disp=0 \else \edef\@tempb{last}\ifx\@tempa\@tempb \global\lock@disp=1 \else \edef\@tempb{all}\ifx\@tempa\@tempb \global\lock@disp=2 \else \global\lock@disp=-1 \fi \fi \fi}
21.1 Choosing the system of lineation

The same questions about where to print the line number apply to sub-lines, and these are the analogous macros for dealing with the problem.

\newcount\sublock@disp
\newcommand{\sublockdisp}[1]{%\l@dgetlock@disp{#1}%
  \ifnum\@l@dtempcntb>\m@ne
    \global\sublock@disp=\@l@dtempcntb
  \else
    \le@warn@BadSublockdisp
  \fi}

We provide a mechanism for using different representations of the line numbers, not just the normal arabic.

\linenumberstyle\linenumrep\linenumr@p\sublinenumberstyle\sublinenumrep\sublinenumr@p

\linenumberstyle and \sublinenumberstyle are user level macros for setting the number representation (\linenumrep and \sublinenumrep) for line and sub-line numbers.

\newcommand*{\linenumberstyle}[1]{%\def\linenumrep##1{\@nameuse{@#1}{##1}}}
\newcommand*{\sublinenumberstyle}[1]{%\def\sublinenumrep##1{\@nameuse{@#1}{##1}}}

Initialise the number styles to arabic.
\linenumberstyle{arabic}
\let\linenumr@p\linenumrep
\sublinenumberstyle{arabic}
\let\sublinenumr@p\sublinenumrep

\leftlinenum\rightlinenum\linenumsep\numlabfont\ledlinenum
\leftlinenum and \rightlinenum are the macros that are called to print marginal line numbers on a page, for left- and right-hand margins respectively.

\linenumsep\numlabfont\ledlinenum
They’re made easy to access and change, since you may often want to change the styling in some way. These standard versions illustrate the general sort of thing that will be needed; they’re based on the \leftheadline macro in The \TeXbook, p. 416.

Whatever these macros output gets printed in a box that will be put into the appropriate margin without any space between it and the line of text. You’ll generally want a kern between a line number and the text, and \linenumsep is provided as a standard way of storing its size. Line numbers are usually printed in a smaller font, and \numlabfont is provided as a standard name for that font. When called, these macros will be executed within a group, so font changes and the like will remain local.

\ledlinenum typesets the line (and subline) number.

The original \numlabfont specification is equivalent to the \LaTeX \scriptsize for a 10pt document.
\newlength{\linenumsep}
21.2 List macros

Reminder: compare these with the \LaTeX list macros in case they would be suitable instead.

We will make heavy use of lists of information, which will be built up and taken apart by the following macros; they are adapted from The \TeXbook, pp. 378–379, which discusses their use in more detail.

These macros consume a large amount of the run-time of this code. We intend to replace them in a future version, and in anticipation of doing so have defined their interface in such a way that it is not sensitive to details of the underlying code.

\list@create The \list@create macro creates a new list. In this version of eledmac this macro doesn’t do anything beyond initializing an empty list macro, but in future versions it may do more.

\list@clear The \list@clear macro just initializes a list to the empty list; in this version of eledmac it is no different from \list@create.

\xright@appenditem \xright@appenditem expands an item and appends it to the right end of a list macro. We want the expansion because we’ll often be using this to store the
current value of a counter. \rightappenditem creates global control sequences, like \def, and uses two temporary token-list registers, \@toksa and \@toksb.

\nevtoks\let\led@toksa=\nevtoks\let\led@toksb
\global\let\led@toksa={\}
\global\let\led@toksb={}
\long\def\rightappenditem#1\rightarrow#2{%
  \global\let\led@toksb=\expandafter{#2}%
  \xdef#2{\the\led@toksb\the\led@toksa\expandafter{#1}}%
  \global\let\led@toksb={}
}
\xleftappenditem \xleftappenditem expands an item and appends it to the left end of a list macro; it is otherwise identical to \rightappenditem.

\gl@p The \gl@p macro removes the leftmost item from a list and places it in a control sequence. You say \gl@p\l\rightarrow\z (where \l is the list macro, and \z receives the left item). \l is assumed nonempty: say \ifx\l\empty to test for an empty \l. The control sequences created by \gl@p are all global.

\def\gl@p#1\rightarrow#2{\expandafter\gl@poff#1\gl@poff#1#2}
\long\def\gl@poff\#1#2\gl@poff#3#4{\gdef#4{#1}\gdef#3{#2}}

21.3 Line-number counters and lists

Footnote references using line numbers rather than symbols can’t be generated in one pass, because we don’t know the line numbers till we ship out the pages. It would be possible if footnotes were never keyed to more than one line; but some footnotes gloss passages that may run for several lines, and they must be tied to the first line of the passage glossed. And even one-line passages require two passes if we want line-per-page numbering rather than line-per-section numbering.

So we run \LaTeX over the text several times, and each time save information about page and line numbers in a ‘line-list file’ to be used during the next pass. At the start of each section—whenever \beginnumbering is executed—the line-list file for that section is read, and the information from it is encoded into a few list macros.

We need first to define the different line numbers that are involved in these macros, and the associated counters.

\line@num The count \line@num stores the line number that’s used in marginal line numbering and in notes: counting either from the start of the page or from the start of the section, depending on your choice for this section. This may be qualified by \subline@num.

\newcount\line@num
The count \subline@num stores a sub-line number that qualifies \line@num. For example, line 10 might have sub-line numbers 1, 2 and 3, which might be printed as lines 10.1, 10.2, 10.3.

\newcount\subline@num

We maintain an associated flag, \ifsublines@, to tell us whether we're within a sub-line range or not.

\sublines@true
\sublines@false

You may wonder why we don't just use the value of \subline@num to determine this—treating anything greater than 0 as an indication that sub-lineation is on. We need a separate flag because sub-lineation can be used together with line-number locking in odd ways: several pieces of a logical line might be interrupted by pieces of sub-lineated text, and those sub-line numbers should not return to zero until the next change in the major line number. This is common in the typesetting of English Renaissance verse drama, in which stage directions are given sub-line numbers: a single line of verse may be interrupted by several stage directions.

\newif\ifsublines@

The count \absline@num stores the absolute number of lines since the start of the section: that is, the number we've actually printed, no matter what numbers we attached to them. This value is never printed on an output page, though \line@num will often be equal to it. It is used internally to keep track of where notes are to appear and where new pages start: using this value rather than \line@num is a lot simpler, because it doesn't depend on the lineation system in use.

\newcount\absline@num

We'll be calling \absline@num numbers 'absolute' numbers, and \line@num and \subline@num numbers 'visible' numbers.

\@lock
\sub@lock

The counts \@lock and \sub@lock tell us the state of line-number and sub-line-number locking. 0 means we're not within a locked set of lines; 1 means we're at the first line in the set; 2, at some intermediate line; and 3, at the last line.

\newcount\@lock
\newcount\sub@lock

Now we can define the list macros that will be created from the line-list file. We will maintain the following lists:

- \line@list: the page and line numbers for every lemma marked by \edtext. There are seven pieces of information, separated by vertical bars:
  1. the starting page,
  2. line, and
  3. sub-line numbers, followed by the
  4. ending page,
  5. line, and
  6. sub-line numbers, and then the
  7. font specifier for the lemma.
These line numbers are all visible numbers. The font specifier is a set of four codes for font encoding, family, series, and shape, separated by / characters. Thus a lemma that started on page 23, line 35 and went on until page 24, line 3 (with no sub-line numbering), and was typeset in a normal roman font would have a line list entry like this:

```
23|35|0|24|3|0|OT1/cmr/m/n.
```

There is one item in this list for every lemma marked by `\edtext`, even if there are several notes to that lemma, or no notes at all. `\edtext` reads the data in this list, making it available for use in the text of notes.

- `\insertlines@list`: the line numbers of lines that have footnotes or other insertions. These are the absolute numbers where the corresponding lemmas begin. This list contains one entry for every footnote in the section; one lemma may contribute no footnotes or many footnotes. This list is used by `\add@inserts` within `\do@line`, to tell it where to insert notes.

- `\actionlines@list`: a list of absolute line numbers at which we are to perform special actions; these actions are specified by the `\actions@list` list defined below.

- `\actions@list`: action codes corresponding to the line numbers in `\actionlines@list`. These codes tell `eledmac` what action it’s supposed to take at each of these lines. One action, the page-start action, is generated behind the scenes by `eledmac` itself; the others, for specifying sub-lineation, line-number locking, and line-number alteration, are generated only by explicit commands in your input file. The page-start and line-number-alteration actions require arguments, to specify the new values for the page or line numbers; instead of storing those arguments in another list, we have chosen the action-code values so that they can encode both the action and the argument in these cases. Action codes greater than $-1000$ are page-start actions, and the code value is the page number; action codes less than $-5000$ specify line numbers, and the code value is a transformed version of the line number; action codes between these two values specify other actions which require no argument.

Here is the full list of action codes and their meanings:

Any number greater than $-1000$ is a page-start action: the line number associated with it is the first line on a page, and the action number is the page number. (The cutoff of $-1000$ is chosen because negative page-number values are used by some macro packages; we assume that page-number values less than $-1000$ are not common.) Page-start action codes are added to the list by the `\page@action` macro, which is (indirectly) triggered by the workings of the `\page@start` macro; that macro should always be called in the output routine, just before the page contents are assembled. `eledmac` calls it in `\pagecontents`.

The action code $-1001$ specifies the start of sub-lineation: meaning that, starting with the next line, we should be advancing `\subline@num` at each start-of-line command, rather than `\line@num`. 
The action code 1002 specifies the end of sub-lineation. At the next start-of-line, we should clear the sub-line counter and start advancing the line number. The action codes for starting and ending sub-lineation are added to the list by the \sub@action macro, as called to implement the \startsub and \endsub macros.

The action code 1003 specifies the start of line number locking. After the number for the current line is computed, it will remain at that value through the next line that has an action code to end locking.

The action code 1004 specifies the end of line number locking.

The action code 1005 specifies the start of sub-line number locking. After the number for the current sub-line is computed, it will remain at that value through the next sub-line that has an action code to end locking.

The action code 1006 specifies the end of sub-line number locking.

The four action codes for line and sub-line number locking are added to the list by the \do@lockon and \do@lockoff macros, as called to implement the \startlock and \endlock macros.

An action code of 5000 or less sets the current visible line number (either the line number or the sub-line number, whichever is currently being advanced) to a specific positive value. The value of the code is -(5000 + n), where n is the value (always ≥ 0) assigned to the current line number. Action codes of this type are added to the list by the \set@line@action macro, as called to implement the \advanceline and \setline macros: this action only occurs when the user has specified some change to the line numbers using those macros. Normally eledmac computes the visible line numbers from the absolute line numbers with reference to the other action codes and the settings they invoke; it doesn’t require an entry in the action-code list for every line.

Here are the commands to create these lists:

\list@create\{line@list\}
\list@create\{insertlines@list\}
\list@create\{actionlines@list\}
\list@create\{actions@list\}

\page@num \endpage@num \endline@num \endsubline@num
\ifnoteschanged@ \noteschanged@true \noteschanged@false

We’ll need some counts while we read the line-list, for the page number and the ending page, line, and sub-line numbers. Some of these will be used again later on, when we are acting on the data in our list macros.

\newcount\page@num \newcount\endpage@num \newcount\endline@num \newcount\endsubline@num

If the number of the footnotes in a section is different from what it was during the last run, or if this is the very first time you’ve run \TeX, on this file, the
information from the line-list used to place the notes will be wrong, and some notes will probably be misplaced. When this happens, we prefer to give a single error message for the whole section rather than messages at every point where we notice the problem, because we don’t really know where in the section notes were added or removed, and the solution in any case is simply to run \LaTeX{} two more times; there’s no fix needed to the document. The \texttt{\textbackslash ifnoteschanged@} flag is set if such a change in the number of notes is discovered at any point.

\begin{verbatim}
\newif\ifnoteschanged@
\newif\ifnoteschanged@
\resetprevline@
\newcommand*{\resetprevline@}{%
\do{\ifnoteschanged@\do{\global\csundef{prevline##1}}%\else\fi}{#1}}%
\dolistloop{\@series}%
\end{verbatim}

Inside the apparatus, at each note, the line number is stored in a macro called \texttt{\textbackslash prevlineX}, where \texttt{X} is the letter of the current series. This macro is called when using \texttt{\textbackslash numberonlyfirstinline}. This macro must be reset at the same time as the line number. The \texttt{\resetprevline@} does this resetting for every series.

\begin{verbatim}
\resetprevline@
\newcommand*{\resetprevline@}{%
\do{\ifcsdef{prevpage##1@num}{\global\csname prevpage##1@num\endcsname=0}{}}%\else\fi}{#1}}%
\dolistloop{\@series}%
\end{verbatim}

Inside the apparatus, at each note, the page number is stored in a macro called \texttt{\textbackslash prevpageX@num}, where \texttt{X} is the letter of the current series. This macro is called when using \texttt{\textbackslash parafootsep}. This macro must be reset at the beginning of each numbered section. The \texttt{\resetprevpage@} command resets this macro for every series.

\begin{verbatim}
\resetprevpage@
\newcommand*{\resetprevpage@}{%
\do{\ifcsdef{prevpage##1@num}{\global\csname prevpage##1@num\endcsname=0}{}}{}%\else\fi}{#1}}%
\dolistloop{\@series}%
\end{verbatim}

\subsection{Reading the line-list file}

\texttt{\textbackslash read@linelist\{file\}} is the control sequence that’s called by \texttt{\begin{numbering}} (via \texttt{\line@list@stuff}) to open and process a line-list file; its argument is the name of the file.

\begin{verbatim}
\newread@inputcheck
\newcommand*{\read@linelist\{\}}{%
\list@clearing@reg
\end{verbatim}

When the file is there we start a new group and make some special definitions we’ll need to process it: it’s a sequence of \TeX{} commands, but they require a few special settings. We make \texttt{[} and \texttt{]} become grouping characters: they’re used that way in the line-list file, because we need to write them out one at a time rather than in balanced pairs, and it’s easier to just use something other than real braces. \texttt{@} must become a letter, since this is run in the ordinary \LaTeX{} context. We
ignore carriage returns, since if we're in horizontal mode they can get interpreted
as spaces to be printed.

Our line, page, and line-locking counters were already zeroed by \line@list@stuff
if this is being called from within \beginnumbering; sub-lineation will be turned
off as well in that case. On the other hand, if this is being called from
\resumenumbering, those things should still have the values they had when
\pausenumbering was executed.

If the file is not there, we print an informative message.
Now, after these preliminaries, we start interpreting the file.

When the reading is done, we're all through with the line-list file. All the
information we needed from it will now be encoded in our list macros.

Finally, we initialize the \next@actionline and \next@action macros, which
specify where and what the next action to be taken is.

This version of \read@linelist creates list macros containing data for the
entire section, so they could get rather large. It would be no more difficult to
read the line-list file incrementally rather than all at once: we could read, at
the start of each paragraph, only the commands relating to that paragraph. But
this would require that we have two line-lists open at once, one for reading, one
for writing, and on systems without version numbers we’d have to do some file
renaming outside of \LaTeX for that to work. We’ve retained this slower approach
to avoid that sort of hacking about, but have provided the \pasenumbering and\resumenumbering macros to help you if you run into macro memory limitations
(see \ref{pausenumbering} p. \pageref{pausenumbering} above).

\section{Commands within the line-list file}

This section defines the commands that can appear within a line-list file. They
all have very short names because we are likely to be writing very large numbers
of them out. One macro, \nl, is especially short, since it will be written to
the line-list file once for every line of text in a numbered section. (Another of
these commands, \lab, will be introduced in a later section, among the cross-referencing commands it is associated with.)

When these commands modify the various page and line counters, they de-
liberately do not say \global. This is because we want them to affect only the
counter values within the current group when nested calls of \@ref occur. (The
code assumes throughout that the value of \globaldefs is zero.)

The macros with \texttt{action} in their names contain all the code that modifies the
action-code list: again, this is so that they can be turned off easily for nested calls
of \@ref.

\texttt{\line@list@version} The \line@list@version check if the line-list file does not refers to the older
commands of \eledmac. In this case, we stop reading the line-list file. Consequently,
\line@list@version should be the first line of a line-number file.

\begin{verbatim}
\newcommand{\line@list@version}[1]{%
  \IfStrEq{#1}{\this@line@list@version}{%
    {}%
  }{%
    \IfLeftRcol%
    \led@warn@Obsolete{\jobname.\extensionchars\the\section@num}%
    \else%
    \led@warn@Obsolete{\jobname.\extensionchars\the\section@num}%
    \fi%
  }%
}\end{verbatim}

\texttt{\nl} \nl does everything related to the start of a new line of numbered text.

\texttt{\nl@reg} In order to get the \setlinenum to work Peter Wilson had to slip in some new
code at the start of the macro, to get the timing of the actions correct. The problem
was that his original naive implementation of \setlinenum had a unfortunate
tendency to change the number of the last line of the \texttt{preceding} paragraph. The
new code is sort of based on the page number handling and \setline It seems
that a lot of fiddling with the line number internals is required.
In November 2004 in order to accurately determine page numbers Peter Wilson added these to the macro. It is now:
\@nl{(page counter number)}\{printed page number\}
I don’t (yet) use the printed number (i.e., the \the\page) but it may come in handy later. The macro \fix\@page checks if a new page has started.

577 \newcommand{\@nl}[2]{%
578 \fix\@page{#1}%
579 \@nl@reg%
580 \newcommand*{\@nl@reg}{%
581 \ıf\l@dchset@num\relax \else
582 \advance\absline@num \@ne
583 \set@line@action
584 \let\l@dchset@num=\relax
585 \advance\absline@num \m@ne
586 \advance\line@num \m@ne
587 \fi

First increment the absolute line-number, and perform deferred actions relating to page starts and sub-lines.
588 \advance\absline@num \@ne
589 \ıf\next@page@num\relax \else
590 \page@action
591 \let\next@page@num=\relax
592 \fi
593 \ıf\sub@change\relax \else
594 \ıf\sub@change>\z@ 595 \sublines@true
596 \else
597 \sublines@false
598 \fi
599 \sub@change=\relax
600 \let\sub@change=\relax
601 \fi

Fix the lock counters, if necessary. A value of 1 is advanced to 2; 3 advances to 0; other values are unchanged.
602 \ıfc@e\@lock
603 \or
604 \@lock \tw@ \or
605 \@lock \z@ \or
606 \fi
607 \ıfc@e\sub@lock
608 \or
609 \sub@lock \tw@ \or
610 \sub@lock \z@ \or
611 \fi
612 \fi

Now advance the visible line number, unless it’s been locked.
21.5 Commands within the line-list file

\ifsynlines\ifnum\nsl<\tw\advance\nslnum \one\fi\else\ifnum\nlock<\tw\advance\linenum \one \slnum \z\fi\fi\\
\fixpage basically replaces \page. It determines whether or not a new page
has been started, based on the page values held by \nl.

\lastpagenum \fixpage \fixpage

\newcount\lastpagenum \lastpagenum=-10000\newcommand*{\fixpage}[1]{\ifnum #1=\lastpagenum\else\ifbypage\csxdef{lastlinenumberon\the\lastpagenum}{\the\linenum}\line\num=\z \slnum=\z\fi\page=\num=\relax\lastpagenum=\num=\relax\def\nextpagenum{#1}\listxadd{\normalpagebreak}{\absline}\fi}}

\@pend These don’t do anything at this point, but will have been added to the auxiliary
\@pendR file(s) if the eledpar package has been used. They are just here to stop eledmac
\@lopL from moaning if the eledpar is used for one run and then not for the following one.
\@lopR

\@subon The \@subon and \@suboff macros turn sub-lineation on and off: but not directly,
\@suboff since such changes don’t really take effect until the next line of text. Instead they
set a flag that notifies \nl of the necessary action.

\newcommand*{\subon}{\ifsynlines\let\subchange=\relax\else\def\subchange{1}\fi}
\newcommand*{\suboff}{\ifsublines\let\subchange=\relax\else\def\subchange{-1}\fi}
\texttt{\@adv} The \texttt{\@adv\{\textit{num}\}} macro advances the current visible line number by the amount specified as its argument. This is used to implement \texttt{\advanceline}.

\begin{verbatim}
\newcommand*{\@adv}[1]{\ifsublines@
  \advance\subline@num by #1\relax
  \ifnum\subline@num<\z@
    \led@warn@BadAdvancelineSubline
    \subline@num \z@
  \fi
  \else
  \advance\line@num by #1\relax
  \ifnum\line@num<\z@
    \led@warn@BadAdvancelineLine
    \line@num \z@
  \fi
  \fi
  \set@line@action}
\end{verbatim}

\texttt{\@set} The \texttt{\@set\{\textit{num}\}} macro sets the current visible line number to the value specified as its argument. This is used to implement \texttt{\setline}.

\begin{verbatim}
\newcommand*{\@set}[1]{\ifsublines@
  \subline@num=#1\relax
  \else
  \line@num=#1\relax
  \fi
  \set@line@action}
\end{verbatim}

\texttt{\l@d@set} The \texttt{\l@d@set\{\textit{num}\}} macro sets the line number for the next \texttt{\pstart...} to the value specified as its argument. This is used to implement \texttt{\setlinenum}. \texttt{\l@dchset@num} is a flag to the \texttt{\@l} macro. If it is not \texttt{\relax} then a linenumber change is to be done.

\begin{verbatim}
\newcommand*{\l@d@set}[1]{%\l@dchset@num
  \line@num=#1\relax
  \def\l@dchset@num{#1}\relax
  \let\l@dchset@num\relax
}\end{verbatim}

\texttt{\page@action} \texttt{\page@action} adds an entry to the action-code list to change the page number.

\begin{verbatim}
\newcommand*{\page@action}{%\page@action
  \xright@appenditem{\the\absline@num to \actionlines@list
  \xright@appenditem{\next@page@num to \actions@list}}
\end{verbatim}
21.5 Commands within the line-list file

\set@line@action \set@line@action adds an entry to the action-code list to change the visible line number.
686 \newcommand*{\set@line@action}{% 
687 \xright@appenditem{\the\absline@num}\to\actionlines@list 
688 \ifs@lines@ 
689 \@l@dtempcnta=-\subline@num 
690 \else 
691 \@l@dtempcnta=-\line@num 
692 \fi 
693 \advance\@l@dtempcnta by -5000 
694 \xright@appenditem{\the\@l@dtempcnta}\to\actions@list}

\sub@action \sub@action adds an entry to the action-code list to turn sub-lineation on or off, according to the current value of the \ifs@lines@ flag.
695 \newcommand*{\sub@action}{% 
696 \xright@appenditem{\the\absline@num}\to\actionlines@list 
697 \ifs@lines@ 
698 \xright@appenditem{-1001}\to\actions@list 
699 \else 
700 \xright@appenditem{-1002}\to\actions@list 
701 \fi}

\lock@on \lock@on adds an entry to the action-code list to turn line number locking on. The current setting of the sub-lineation flag tells us whether this applies to line numbers or sub-line numbers.
Adding commands to the action list is slow, and it’s very often the case that a lock-on command is immediately followed by a lock-off command in the line-list file, and therefore really does nothing. We use a look-ahead scheme here to detect such pairs, and add nothing to the line-list in those cases.
702 \newcommand*{\lock@on}{\futurelet\next\do@lockon}
703 \newcommand*{\do@lockon}{% 
704 \ifx\next\lock@off 
705 \global\let\lock@off=\skip@lockoff 
706 \else 
707 \do@lockonL 
708 \fi 
709 \newcommand*{\do@lockonL}{% 
710 \xright@appenditem{\the\absline@num}\to\actionlines@list 
711 \ifs@lines@ 
712 \xright@appenditem{-1005}\to\actions@list 
713 \ifnum\sub@lock=\z@ 
714 \sub@lock \@ne 
715 \else 
716 \ifnum\sub@lock=\thr@@ 
717 \sub@lock \@ne 
718 \fi 
719 \fi 
720 \else 
721 \xright@appenditem{-1003}\to\actions@list
\lock@off \lock@off adds an entry to the action-code list to turn line number locking off.
\do@lockoff \do@lockoffL
\skip@lockoff
These macros implement the \makeindex command. They use a new action code, namely 1007.
\n@num
\n@num@stanza
This macro implements the \makeindex for stanza command. It uses a new action code, namely 1008.
21.5 Commands within the line-list file

\ifledRcol%  \
xright@appenditem{\the\absline@numR}\to\actionlines@listR%  \
xright@appenditem{-1008}\to\actions@listR%  \
\else%  \
xright@appenditem{\the\absline@num}\to\actionlines@list%%  \
xright@appenditem{-1008}\to\actions@list%  \
\fi%  
\}

\ifl@dhidenumber  \hidenumbering hides number in margin. It uses action code 1009.
\hidenumbering
\hides number in margin. It uses action code 1009.
\h@num
\newif\ifl@dhidenumber
\newcommand*{\hidenumbering}{  \
\ifledRcol%  \
\write\linenum@outR{\string\hide@num}%  \
\else%  \
\write\linenum@out{\string\hide@num}%  \
\fi%  
}\newcommand*{\hide@num}{  \
\ifledRcol%  \
xright@appenditem{\the\absline@numR}\to\actionlines@listR%  \
xright@appenditem{-1009}\to\actions@listR%  \
\else%  \
xright@appenditem{\the\absline@num}\to\actionlines@list%%  \
xright@appenditem{-1009}\to\actions@list%  \
\fi%  
}\%
\newcommand*{\hide@num}{%  
\ifledRcol%  \
xright@appenditem{\the\absline@numR}\to\actionlines@listR%  \
xright@appenditem{-1009}\to\actions@listR%  \
\else%  \
xright@appenditem{\the\absline@num}\to\actionlines@list%%  \
xright@appenditem{-1009}\to\actions@list%  \
\fi%  
}\%
\newcommand*{\hide@num}{%  
\ifledRcol%  \
xright@appenditem{\the\absline@numR}\to\actionlines@listR%  \
xright@appenditem{-1009}\to\actions@listR%  \
\else%  \
xright@appenditem{\the\absline@num}\to\actionlines@list%%  \
xright@appenditem{-1009}\to\actions@list%  \
\fi%  
}\%
\newcommand*{\hide@num}{%  
\ifledRcol%  \
xright@appenditem{\the\absline@numR}\to\actionlines@listR%  \
xright@appenditem{-1009}\to\actions@listR%  \
\else%  \
xright@appenditem{\the\absline@num}\to\actionlines@list%%  \
xright@appenditem{-1009}\to\actions@list%  \
\fi%  
}\%
\newcommand*{\hide@num}{%  
\ifledRcol%  \
xright@appenditem{\the\absline@numR}\to\actionlines@listR%  \
xright@appenditem{-1009}\to\actions@listR%  \
\else%  \
xright@appenditem{\the\absline@num}\to\actionlines@list%%  \
xright@appenditem{-1009}\to\actions@list%  \
\fi%  
}\%
\newcommand*{\hide@num}{%  
\ifledRcol%  \
xright@appenditem{\the\absline@numR}\to\actionlines@listR%  \
xright@appenditem{-1009}\to\actions@listR%  \
\else%  \
xright@appenditem{\the\absline@num}\to\actionlines@list%%  \
xright@appenditem{-1009}\to\actions@list%  \
\fi%  
}\%
\newcommand*{\hide@num}{%  
\ifledRcol%  \
xright@appenditem{\the\absline@numR}\to\actionlines@listR%  \
xright@appenditem{-1009}\to\actions@listR%  \
\else%  \
xright@appenditem{\the\absline@num}\to\actionlines@list%%  \
xright@appenditem{-1009}\to\actions@list%  \
\fi%  
}\%
\newcommand*{\hide@num}{%  
\ifledRcol%  \
xright@appenditem{\the\absline@numR}\to\actionlines@listR%  \
xright@appenditem{-1009}\to\actions@listR%  \
\else%  \
xright@appenditem{\the\absline@num}\to\actionlines@list%%  \
xright@appenditem{-1009}\to\actions@list%  \
\fi%  
}\%
\newcommand*{\hide@num}{%  
\ifledRcol%  \
xright@appenditem{\the\absline@numR}\to\actionlines@listR%  \
xright@appenditem{-1009}\to\actions@listR%  \
\else%  \
xright@appenditem{\the\absline@num}\to\actionlines@list%%  \
xright@appenditem{-1009}\to\actions@list%  \
\fi%  
}\%
\newcommand*{\hide@num}{%  
\ifledRcol%  \
xright@appenditem{\the\absline@numR}\to\actionlines@listR%  \
xright@appenditem{-1009}\to\actions@listR%  \
\else%  \
xright@appenditem{\the\absline@num}\to\actionlines@list%%  \
xright@appenditem{-1009}\to\actions@list%  \
\fi%  
}\%
\newcommand*{\hide@num}{%  
\ifledRcol%  \
xright@appenditem{\the\absline@numR}\to\actionlines@listR%  \
xright@appenditem{-1009}\to\actions@listR%  \
\else%  \
xright@appenditem{\the\absline@num}\to\actionlines@list%%  \
xright@appenditem{-1009}\to\actions@list%  \
\fi%  
}\%}
\@ref  \Dummy@ref \@ref marks the start of a passage, for creation of a footnote reference. It takes two arguments:
• #1, the number of entries to add to \insertlines@list for this reference. This value, here and within \edtext, which computes it and writes it to the line-list file, will be stored in the count \insert@count.
\newcount\insert@count

• #2, a sequence of other line-list-file commands, executed to determine the ending line-number. (This may also include other \@ref commands, corresponding to uses of \edtext within the first argument of another instance of \edtext.)
\newcommand*{\dummy@ref}{#2}
\@ref@reg
\Dummy@ref When nesting of \@ref commands does occur, it’s necessary to temporarily redefine \@ref within \@ref, so that we’re only doing one of these at a time.
\newcommand*{\dummy@ref}{#2}
\@ref@reg
\Dummy@ref The first thing \@ref (i.e. \@ref@reg) itself does is to add the specified number of items to the \insertlines@list list.
Next, process the second argument to determine the page and line numbers for the end of this lemma. We temporarily equate \@ref to a different macro that just executes its argument, so that nested \@ref commands are just skipped this time. Some other macros need to be temporarily redefined to suppress their action.

\begin{group}
\let\@ref=\dummy@ref
\let@lopL\@gobble
\let\page@action=\relax
\let\sub@action=\relax
\let\set@line@action=\relax
\let\@lab=\relax
\let\@lemma=\relax%
\let\@sw=\@gobblethree%
#2
\global\endpage@num=\page@num
\global\endline@num=\line@num
\global\endsubline@num=\subline@num
\endgroup

Now store all the information about the location of the lemma's start and end in \line@list.
\xright@appenditem%
\let\@page@num=\the\line@num%
\ifsplines@ \the\subline@num \else 0 \fi%
\let\@endpage@num=\the\endline@num%
\ifsplines@ \the\endsubline@num \else 0 \fi\to\line@list
Create a list which stores every second argument of each \@sw in this lemma, at this level. Also set the boolean about the use of lemma in this edtext level to false.
\expandafter\list@create\expandafter{\csname sw@list@edtext@tmp@\the\@edtext@level\endcsname}
\providebool{lemmacommand@\the\@edtext@level}
\boolfalse{lemmacommand@\the\@edtext@level}

Execute the second argument of \@ref again, to perform for real all the commands within it.
#2%

Now, we store the list of \@sw of this current \edtext as an element of the global list of list of \@sw for a \edtext depth.
\ifnum\@edtext@level>0%
\def\create@this@edtext@level\expandafter{\expandafter{\expandafter{\csname sw@list@edtext@tmp@\the\@edtext@level\endcsname}}}
\else
\def\create@this@edtext@level{\expandafter{\expandafter{\expandafter{\csname sw@list@edtext@tmp@\the\@edtext@level\endcsname}}}}
\fi
21.6 Writing to the line-list file

We’ve now defined all the counters, lists, and commands involved in reading the line-list file at the start of a section. Now we’ll cover the commands that *eledmac* uses within the text of a section to write commands out to the line-list.

The file will be opened on output stream `\linenum@out`.

Once any file is opened on this stream, we keep it open forever, or else switch to another file that we keep open. The reason is that we want the output routine to write the page number for every page to this file; otherwise we’d have to write it at the start of every line. But it’s not very easy for the output routine to tell whether an output stream is open or not. There’s no way to test the status of a particular output stream directly, and the asynchronous nature of output routines makes the status hard to determine by other means.

We can manage pretty well by means of the `\iffirst@linenum@out@` flag; its inelegant name suggests the nature of the problem that made its creation necessary. It’s set to be `true` before any `\linenum@out` file is opened. When such a file is opened for the first time, it’s done using `\immediate`, so that it will at once be safe for the output routine to write to it; we then set this flag to `false`.

The commands allowed in the line-list file and their arguments can change between two version of *eledmac*. The `\this@line@list@version` command is upgraded when it happens. It is written in the file list. If we process a line-list file which used a older version, that means the commands used insided are deprecated, and we can’t use them.

The `\line@list@stuff{⟨file⟩}` macro, which is called by `\beginnumbering`, performs all the line-list operations needed at the start of a section. Its argument is the name of the line-list file.
First, use the commands of the previous section to interpret the line-list file from the last run.
\read@linelist{#1}%

Now close the current output line-list file, if any, and open a new one. The first time we open a line-list file for output, we do it using \immediate, and clear the \iffirst@linenum@out@ flag.
\iffirst@linenum@out@
  \immediate\closeout\linenum@out%
  \global\first@linenum@out@false%
  \immediate\openout\linenum@out=#1\relax%
  \immediate\write\linenum@out{\string\line@list@version{\this@line@list@version}}%
\else
  If we get here, then this is not the first line-list we've seen, so we don't open or close the files immediately.
  \if@minipage%
    \leavevmode%
  \fi%
  \closeout\linenum@out%
  \openout\linenum@out=#1\relax%
  \fi}

\new@line The \new@line macro sends the \@nl command to the line-list file, to mark the start of a new text line, and its page number.
\newcommand*{\new@line}{%
\if@noneed@Footnote is a boolean to check if we have to print an error message when a \texttt{\edtext} is called without any footnotes.

\flag@start We enclose a lemma marked by \texttt{\edtext} in \texttt{\flag@start} and \texttt{\flag@end}: these send the \texttt{\@ref} command to the line-list file. \texttt{\edtext} is responsible for setting the value of \texttt{\insert@count} appropriately; it actually gets done by the various footnote macros.

\newcommand*{\flag@start}{\if\fieldRcol\def\next{\write\linenum@outR{\string\@ref[\the\insert@countR]}}\else\ifnum\insert@countR<1\if@noneed@Footnote\else\led@err@EdtextWithoutFootnote\fi\else\edef\next{\write\linenum@out{\string\@ref[\the\insert@count]}}\fi\fi\else\edef\next{\write\linenum@out{\string\@ref[\the\insert@count]}}\fi\fi}

\page@start Originally the commentary was: \texttt{\page@start} writes a command to the line-list file noting the current page number; when used within an output routine, this should be called so as to place its \texttt{\write} within the box that gets shipped out, and as close to the top of that box as possible.

However, in October 2004 Alexej Krukov discovered that when processing long paragraphs that included Russian, Greek and Latin texts \texttt{eledmac} would go into an infinite loop, emitting thousands of blank pages. This was caused by being unable to find an appropriate place in the output routine. A different algorithm is now used for getting page numbers.

\newcommand*{\page@start}{}

\startsub and \endsub turn sub-lineation on and off, by writing appropriate instructions to the line-list file. When sub-lineation is in effect, the line number
counter is frozen and the sub-line counter advances instead. If one of these commands appears in the middle of a line, it doesn’t take effect until the next line; in other words, a line is counted as a line or sub-line depending on what it started out as, even if that changes in the middle.

We tinker with \lastskip because a command of either sort really needs to be attached to the last word preceding the change, not the first word that follows the change. This is because sub-lineation will often turn on and off in mid-line—stage directions, for example, often are mixed with dialogue in that way—and when a line is mixed we want to label it using the system that was in effect at its start. But when sub-lineation begins at the very start of a line we have a problem, if we don’t put in this code.

\[\text{\newcommand*{\startsub}{\dimen0\lastskip\ifdim\dimen0>0pt \unskip \fi\write\linenum@out{\string\sub@on} \ifdim\dimen0>0pt \hskip\dimen0 \fi}}\]

\[\text{\def\endsub{\dimen0\lastskip\ifdim\dimen0>0pt \unskip \fi\write\linenum@out{\string\sub@off} \ifdim\dimen0>0pt \hskip\dimen0 \fi}}\]

\advanceline You can use \advanceline{(num)} in running text to advance the current visible line-number by a specified value, positive or negative.

\[\text{\newcommand*{\advanceline}{[1]}{\write\linenum@out{\string\@adv[#1]}}}\]

\setline You can use \setline{(num)} in running text (i.e., within \pstart...\pend) to set the current visible line-number to a specified positive value.

\[\text{\newcommand*{\setline}{[1]}%\ifnum#1<\z@ \led@warn@BadSetline \else \write\linenum@out{\string\@set[#1]}% \fi}\]

\setlinenum You can use \setlinenum{(num)} before a \pstart to set the visible line-number to a specified positive value. It writes a \l@d@set command to the line-list file.

\[\text{\newcommand*{\setlinenum}{[1]}%\ifnum#1<\z@ \led@warn@BadSetlinenum \else \write\linenum@out{\string\l@d@set[#1]}% \fi}\]

\startlock \endlock You can use \startlock or \endlock in running text to start or end line number locking at the current line. They decide whether line numbers or sub-line numbers are affected, depending on the current state of the sub-lineation flags.
22 Marking text for notes

The \edtext (or \critext) macro is used to create all footnotes and endnotes, as well as to print the portion of the main text to which a given note or notes is keyed. The idea is to have that lemma appear only once in the .tex file: all instances of it in the main text and in the notes are copied from that one appearance.

For convenience, I will use \*text when I do not need to distinguish between \edtext and \critext. The \*text macros take two arguments, the only difference between \edtext and \critext is how the second argument is delineated.

\critext requires two arguments. At any point within numbered text, you use it by saying:

\critext{#1}{#2} \\

Similarly \edtext requires the same two arguments but you use it by saying:

\edtext{#1}{#2} \\

• #1 is the piece of the main text being glossed; it gets added to the main text, and is also used as a lemma for notes to it.
• \#2 is a series of subsidiary macros that generate various kinds of notes. With \critext the / after \#2 must appear: it marks the end of the macro. (The \TeXbook, p. 204, points out that when additional text to be matched follows the arguments like this, spaces following the macro are not skipped, which is very desirable since this macro will never be used except within text. Having an explicit terminator also helps keep things straight when nested calls to \critext are used.) Braces around \#2 are optional with \critext and required for \edtext.

The \*text macro may be used (somewhat) recursively; that is, \*text may be used within its own first argument. The code would be much simpler without this feature, but nested notes will commonly be necessary: it’s quite likely that we’ll have an explanatory note for a long passage and notes on variants for individual words within that passage. The situation we can’t handle is overlapping notes that aren’t nested: for example, one note covering lines 10–15, and another covering 12–18. You can handle such cases by using the \lemma and \linenum macros within \#2: they alter the copy of the lemma and the line numbers that are passed to the notes, and hence allow you to overcome any limitations of this system, albeit with extra effort.

The recursive operation of \*text will fail if you try to use a copy that is called something other than \*text. In order to handle recursion, \*text needs to redefine its own definition temporarily at one point, and that doesn’t work if the macro you are calling is not actually named \*text. There’s no problem as long as \*text is not invoked in the first argument. If you want to call \*text something else, it is best to create instead a macro that expands to an invocation of \*text, rather than copying \*text and giving it a new name; otherwise you will need to add an appropriate definition for your new macro to \morenoexpands.

Side effects of our line-numbering code make it impossible to use the usual footnote macros directly within a paragraph whose lines are numbered (see comments to \do@line, \[\text{\text{p. 107}}\]). Instead, the appropriate note-generating command is appended to the list macro \inserts@list, and when \pend completes the paragraph it inserts all the notes at the proper places.

Note that we don’t provide previous-note information, although it’s often wanted; your own macros must handle that. We can’t do it correctly without keeping track of what kind of notes have gone past: it’s not just a matter of remembering the line numbers associated with the previous invocation of \*text, because that might have been for a different kind of note. It is preferable for your footnote macros to store and recall this kind of information if they need it.

22.1 \edtext (and \critext) itself

The various note-generating macros might want to request that commands be executed not at once, but in close connection with the start or end of the lemma. For example, footnote numbers in the text should be connected to the end of the lemma; or, instead of a single macro to create a note listing variants, you might want to use several macros in series to create individual variants, which would
each add information to a private macro or token register, which in turn would be formatted and output when all of \#2 for the lemma has been read.

To accommodate this, we provide a list macro to which macros may add commands that should subsequently be executed at the end of the lemma when that lemma is added to the text of the paragraph. A macro should add its contribution to \end@lemmas by using \xleft@appenditem. (Anything that needs to be done at the start of the lemma may be handled using \aftergroup, since the commands specified within \edtext’s second argument are executed within a group that ends just before the lemma is added to the main text.)

\end@lemmas is intended for the few things that need to be associated with the end of the lemma, like footnote numbers. Such numbers are not implemented in the current version, and indeed no use is currently made of \end@lemmas or of the \aftergroup trick. The general approach would be to define a macro to be used within the second argument of \edtext that would add the appropriate command to \end@lemmas.

Commands that are added to this list should always take care not to do anything that adds possible line-breaks to the output; otherwise line numbering could be thrown off.

We now need to define a number of macros that allow us to weed out nested instances of \edtext, and other problematic macros, from our lemma. This is similar to what we did in reading the line-list file using \dummy@ref and various redefinitions—and that’s because nested \edtext macros create nested \@ref entries in the line-list file.

Here’s a macro that takes the same arguments as \critext but merely returns the first argument and ignores the second.

\begin{verbatim}
\long\def\dummy@text#1#2/{#1}
\end{verbatim}

\def\dummy@edtext We now need to define a number of macros that allow us to weed out nested instances of \edtext, and other problematic macros, from our lemma. This is similar to what we did in reading the line-list file using \dummy@ref and various redefinitions—and that’s because nested \edtext macros create nested \@ref entries in the line-list file.

Here’s a macro that takes the same arguments as \critext but merely returns the first argument and ignores the second.

\begin{verbatim}
\long\def\dummy@text#1#2/{#1}
\end{verbatim}

\def\dummy@edtext Is\TeX\ users are not used to delimited arguments, so we provide a \edtext macro as well.

\newcommand{\dummy@edtext}[2][2]{\#1}

\def\dummy@edtext Some time, we want to obtain only the first argument of \edtext, while also wrapping it in \showlemma. For example, when printing a \eledsection.

\newcommand{\dummy@edtext@showlemma}{\#1}

We’re going to need another macro that takes one argument and ignores it entirely. This is supplied by the Is\TeX \@gobble{⟨arg⟩}.

\begin{verbatim}
\no@expands\morenoexpands
\end{verbatim}

We need to turn off macro expansion for certain sorts of macros we’re likely to see within the lemma and within the notes.

The first class is font-changing macros. We suppress expansion for them by letting them become equal to zero.\footnote{Since ‘control sequences equivalent to characters are not expandable’—The \TeX\book, answer to Exercise 20.14.} This is done because we want to pass into
our notes the generic commands to change to roman or whatever, and not their expansions that will ask for a particular style at a specified size. The notes may well be in a smaller font, so the command should be expanded later, when the note’s environment is in effect.

A second sort to turn off includes a few of the accent macros. Most are not a problem: an accent that’s expanded to an \accent command may be harder to read but it works just the same. The ones that cause problems are: those that use alignments—\TeX seems to get confused about the difference between alignment parameters and macro parameters; those that use temporary control sequences; and those that look carefully at what the current font is.

(The \copyright macro defined in Plain \TeX has this sort of problem as well, but isn’t used enough to bother with. That macro, and any other that causes trouble, will get by all right if you put a \protect in front of it in your file.)

We also need to eliminate all \eledmac macros like \edlabel and \setline that write things to auxiliary files: that writing should be done only once. And we make \edtext itself, if it appears within its own argument, do nothing but copy its first argument.

Finally, we execute \morenoexpands. The version of \morenoexpands defined here does nothing; but you may define a version of your own when you need to add more expansion suppressions as needed with your macros. That makes it possible to make such additions without needing to copy or modify the standard \eledmac code. If you define your own \morenoexpands, you must be very careful about spaces: if the macro adds any spaces to the text when it runs, extra space will appear in the main text when \edtext is used.

(A related problem, not addressed by these two macros, is that of characters whose category code is changed by any the macros used in the arguments to \edtext. Since the category codes are set when the arguments are scanned, macros that depend on changing them will not work. We have most often encountered this with characters that are made ‘active’ within text in some, but not all, of the languages used within the document. One way around the problem, if it takes this form, is to ensure that those characters are always active; within languages that make no special use of them, their associated control sequences should simply return the proper character.)

\newcommand*{\no@expands}{%
\let\select@@lemmafont=0%
\let\startsub=\relax \let\endsub=\relax
\let\startlock=\relax \let\endlock=\relax
\let\edlabel=\@gobble
\let\setline=\@gobble \let\endlock=\relax
\let\edtext=\dummy@edtext
\let\critext=\dummy@text
\let\same\@gobble \let\advanceline=\@gobble
\let\morenoexpands=\relax
\let\advanceline=\@gobble
\let\morenoexpands=\relax
}
\edtext (and \critext) itself

\@tag  Now, we define an empty \@tag command. It will be redefine by \edtext: its value is the first args. It will be used by the \Xfootnote commands.

\newcommand{\@tag}{}

\@edtext@level  This counter is increased by 1 at each level of \edtext (or \critext). That is useful for some commands which can have a different behavior if called inside or outside of the \{\textit{lemma}\} argument.

\newcount\@edtext@level\%\@edtext@level=0\%

\critext  Now we begin \critext itself. The definition requires a / after the arguments: this eliminates the possibility of problems about knowing where #2 ends. This also changes the handling of spaces following an invocation of the macro: normally such spaces are skipped, but in this case they’re significant because #2 is a ‘delimited parameter’. Since \critext is always used in running text, it seems more appropriate to pay attention to spaces than to skip them.

Since v.1.17.0, \critext only refers to \edtext.

\long\def\critext#1#2/{\edtext{#1}{#2}}\%

\edtext  When executed, \edtext first ensures that we’re in horizontal mode.

\newcommand{\edtext}[2]{\leavevmode\%
Then, check if we are in a numbered paragraph (\pstart…\pend).\%
\ifnumberedpar\%
We increase the \@edtext@ counter to know in which level of \edtext we are.\%
\global\advance\@edtext@level by 1\%
By default, we don’t use \textit{lemma} \%
\global\@lemmacommand@false\%
\begingroup\%
We get the next series of samewords data in the list of samewords data for the current edtext level. We push them inside \texttt{sw@inthisedtext}.\%
\ifledRcol\%
\ifcsundef{sw@list@edtextR@\the\@edtext@level}\%
{\global\let\sw@inthisedtext\empty}\%
\else\%
\ifcsundef{sw@list@edtext@\the\@edtext@level}\%
{\global\let\sw@inthisedtext\empty}\%
\fi\%
\fi\%
\fi\edtextRcol\%
@tag  Our normal lemma is just argument \#1; but that argument could have further
invocations of \texttt{edtext} within it. We get a copy of the lemma without any \texttt{edtext}
macros within it by temporarily redefining \texttt{edtext} to just copy its first argument
and ignore the other, and then expand \#1 into @tag, our lemma.

This is done within a group that starts here, in order to get the original \texttt{edtext}
restored; within this group we’ve also turned off the expansion of those control
sequences commonly found within text that can cause trouble for us.

\begin{verbatim}
\global\renewcommand{@tag}{%
   \no@expands \#1%
}\end{verbatim}

@l@d@nums  Prepare more data for the benefit of note-generating macros: the line references
and font specifier for this lemma go to @l@d@nums.

\begin{verbatim}
\set@line%
\insert@count
\end{verbatim}

\texttt{insert@count} will be altered by the note-generating macros: it counts the
number of deferred footnotes or other insertions generated by this instance of
\texttt{edtext}. If we are in a right column (eledpar), we use \texttt{insert@countR} instead
of \texttt{insert@count}.

\begin{verbatim}
\ifledRcol \global\insert@countR \z@%
\else \global\insert@count \z@ \fi%
\end{verbatim}

Now process the note-generating macros in argument \#2 (i.e., \texttt{Afootnote}, \texttt{lemma},
\texttt{etc.}). \texttt{ignorespaces} is here to skip over any spaces that might appear at the
start of \#2; otherwise they wind up in the main text. Footnote and other macros
that are used within \#2 should all end with \texttt{ignorespaces} as well, to skip any
spaces between macros when several are used in series.

\begin{verbatim}
\ignorespaces \#2\relax%
\end{verbatim}

With polyglossia, you must track whether the language reads left to right (English)
or right to left (Arabic).

\begin{verbatim}
\@ifundefined{xpg@main@language}{%if not polyglossia
   \flag@start}%
\if@RTL\flag@end\else\flag@start\fi%
\end{verbatim}

We write in the numbered file whether the current \texttt{edtext} has a \texttt{lemma} in the
the second argument.

\begin{verbatim}
\if@lemmacommand0%
\ifledRcol1%
\write\linenum@outR{\string\@lemma}%
\else%
\write\linenum@out{\string\@lemma}%
\fi%
\fi%
\end{verbatim}

Finally, we’re ready to admit the first argument into the current paragraph.

It’s important that we generate and output all the notes for this chunk of
text \emph{before} putting the text into the paragraph: notes that are referenced by line
number should generally be tied to the start of the passage they gloss, not the end. That should all be done within the expansion of \texttt{\#2} above, or in \texttt{\aftergroup} commands within that expansion.

Finally, we add any insertions that are associated with the end of the lemma. Footnotes that are identified by symbols rather than by where the lemma begins in the main text need to be done here, and not above.

We switch to false some flags.

- The one that checks having footnotes inside a \texttt{\edtext}.
- The one that says we are inside a \texttt{\edtext}.
- The one that says we are inside à \texttt{\@lemma}.

If we are outside of a numbered paragraph, we send error message and print the first argument.

\texttt{\ifnumberline} The \texttt{\ifnumberline} option can be set to FALSE to disable line numbering.

\texttt{\newif\ifnumberline \numberlinetrue}
\set@line  The \set@line macro is called by \critext to put the line-reference field and
font specifier for the current block of text into \l@d@nums.

One instance of \critext may generate several notes, or it may generate
none—it’s legitimate for argument #2 to \critext to be empty. But \flag@start
and \flag@end induce the generation of a single entry in \line@list during the
next run, and it’s vital to also remove one and only one \line@list entry here.

\newcommand*{\set@line}{%  

  If no more lines are listed in \line@list, something’s wrong—probably just
  some change in the input. We set all the numbers to zeros, following an old
  publishing convention for numerical references that haven’t yet been resolved.

  \ifx\line@list\empty
  \global\noteschanged@true
  \xdef\l@d@nums{000|000|000|000|000|000|%\edfont@info}
  \else
  \gl@p\line@list\to\@tempb
  \xdef\l@d@nums{\@tempb|\edfont@info}|%\global\let\@tempb=\undefined
  \fi

\edfont@info  The macro \edfont@info returns coded information about the current font.

\newcommand*{\edfont@info}{\f@encoding/\f@family/\f@series/\f@shape}

22.2 Substitute lemma

\lemma  The \lemma{⟨text⟩} macro allows you to change the lemma that’s passed on to
the notes. Read about \@tag in normal \edtext macro for more details about
sw@list@inedtext and \no@expands \[22.1\] p. \[44].

\newcommand*{\lemma}[1]{%  
  \global\@lemmacommand@true%
  \global\renewcommand{\@tag}{%  
    \no@expands #1%
  }%
  \ignorespaces%
}

\if@lemmacommand%  This boolean is set to TRUE inside a \edtext (or \critext) when a \lemma
command is called. That is useful for some commands which can have a different
behavior if the lemma in the note is different from the lemma in the main text.

\newif\if@lemmacommand%

\@lemma  The \@lemma is written in the numbered file to set which \edtext has an \lemma
as second argument.

\newcommand{\@lemma}{%  
  \booltrue{lemmacommand\the@edtext@level}%
}

\if@lemmacommand%
22.3 Substitute line numbers

\texttt{\texttt{linenum}} The \texttt{linenum} macro can change any or all of the page and line numbers that are passed on to the notes.

As argument \texttt{linenum} takes a set of seven parameters separated by vertical bars, in the format used internally for \texttt{\@d@nums} (see 21.3 p. 72): the starting page, line, and sub-line numbers, followed by the ending page, line, and sub-line numbers, and then the font specifier for the lemma. However, you can omit any parameters you don’t want to change, and you can omit a string of vertical bars at the end of the argument. Hence \texttt{linenum{18|4|0|18|7|1|0}} is an invocation that changes all the parameters, but \texttt{linenum{|3}} only changes the starting line number, and leaves the rest unaltered.

We use \texttt{\textbackslash{} as an internal separator for the macro parameters.}
1061 \texttt{\newcommand*{\linenum}{[1]{\xdef\@tempa{#1\\|\\|\\|\\|\\|\\|\noexpand\\l@d@nums}\global\let\l@d@nums=\empty\expandafter\line@set\@tempa|\ignorespaces}}}
\texttt{\line@set} \texttt{linenum} calls \texttt{\line@set} to do the actual work; it looks at the first number in the argument to \texttt{linenum}, sets the corresponding value in \texttt{\@d@nums}, and then calls itself to process the next number in the \texttt{linenum} argument, if there are more numbers in \texttt{\@d@nums} to process.
1065 \texttt{\def\line@set{[1]{#1|#2\|#3\|#4\|\\\\\\\\\\\}}}\texttt{\%}
1066 \texttt{\gdef\@tempb{#1}\%}
1067 \texttt{\textbackslash{if}\x@tempb\textbackslash{empty}\%}
1068 \texttt{\textbackslash{\@d@add{#3}\\%}}
1069 \texttt{\textbackslash{else}}
1070 \texttt{\textbackslash{\@d@add{#1}\\%}}
1071 \texttt{\textbackslash{fi}}
1072 \texttt{\gdef\@tempb{#4}\%}
1073 \texttt{\textbackslash{if}\x@tempb\textbackslash{empty}\textbackslash{else}}
1074 \texttt{\textbackslash{\@d@add{}}\line@set{#2\\#4\\\\\\\\\\\\\\\\\\\\\\\%}}
1075 \texttt{\textbackslash{fi}}

\texttt{\@d@add} \texttt{\line@set} uses \texttt{\@d@add} to tack numbers or vertical bars onto the right hand end of \texttt{\@d@nums}.
1076 \texttt{\newcommand{\@d@add}[1]{\xdef\@d@nums{\@d@nums#1}}}\texttt{\%}

22.4 Lemma disambiguation

The mechanism which counts the occurrence of a same word in a same line is quite complex, because, when \LaTeX reads a command between a \texttt{pstart} and a \texttt{pend}, it does not know yet which are the line numbers.

The general mechanism is the following:

- \textbf{At the first run}, each \texttt{\textbackslash{sameword}} command increments an \texttt{etoolbox} counter the name of which contains the argument of the \texttt{\textbackslash{sameword}} commands.
• Then this counter, associated with the argument of `\sameword` is stored, with the `\sw` command, in the auxiliary file of the current `eledmac` section (the `.1`, `.2`... file).

• **When this auxiliary file is read at the second run**, different operations are achieved:

1. Get the rank of each `\sameword` in a line (relative rank) from the rank of each `\sameword` in all the numbered section (absolute rank):
   - For each paired `\sameword` argument and absolute line number, a counter is defined. Its value corresponds to the number of times `\sameword{argument}` is called from the beginning of the lineation to the end of the current line. We also store the same data for the preceding absolute line number, if it does not have `\sameword{argument}`.
   - For each `\sameword` having the same argument, we substract from its absolute rank the number stored for the paired `\sameword` argument and previous absolute line number. Consequently, we obtain the relative rank.
   - See the following example which explain how for same `\sameword` absolute ranks are transformed to relative rank.

   At line 1:
   - absolute rank 1 becomes relative rank 1-0 = 1
   - 1 is stored for this `\sameword` and the line 1

   At line 2:
   - absolute rank 2 becomes relative rank 2-1 = 1
   - absolute rank 3 becomes relative rank 3-2 = 2
   - 3 is stored for this `\sameword` and the line 2

   At line 3:
   - no `\sameword` for this line.
   - 3 is stored for this `\sameword` and the line 3

   At line 4:
   - absolute rank 4 becomes relative rank 4-3 = 1
   - 3 is stored for this `\sameword` and the line 4

2. Create lists of lists of `\sameword` by depth of `\edtext`. That is: create a list for `\edtext` of level 1, a list for `\edtext` of level 2, a list for `\edtext` of level 3 etc. For each `\edtext` in these list, we store all the relative rank of `\sameword` which are called as lemma information, that is 1) or called in the first argument of `\sameword` 2) or called in the `\lemma` macro of the second argument of `\sameword` AND marked by the optional argument of `\sameword` in first argument of `\edtext`.
   For example, suppose a line with nested `\edtext`s which contains some word marked by `\sameword` and having the following relative rank:
22.4 Lemma disambiguation

In this example, all lemma information for \texttt{\edtext} is framed. The text in parenthesis is the content of critical notes associated to the preceding frame. As you can see, we have two level of \texttt{\edtext}.

The list for \texttt{\edtext}s of level 1 is \{1, 2, 2, 3, 4, 3\}, \{5, 4\}.

The list for \texttt{\edtext}s of level 2 is \{1, 2, 2, 3\}, \{5\}.

As you can see, the mandatory argument of \texttt{\sameword} does not matter: we store the rank informations for every word potentially ambiguous.

- At the second run, when a critical notes is called, we associate it to the next item of the list associated to is \texttt{\edtext} level. So, in the previous example:
  - Critical notes (A) and (B) are associated with \{1, 2, 2, 3\}.
  - Critical note (C) is associated with \{1, 2, 2, 3, 4, 3\}.
  - Critical note (D) is associated with \{5\}.
  - Critical note (E) is associated with \{5, 4\}.

- At the second run, when a critical note is printed:
  - The \texttt{\sameword} command is let \texttt{\sameword@inedtext}.
  - At each call of this \texttt{\sameword@inedtext}, we step to the next element of the list associated to the note. Let it be \texttt{r}.
  - For the word marked by \texttt{\sameword}, we calculate how many time it is called in its line. To do it:
    * We get the absolute line number of the current \texttt{\sameword}. This absolute line number was stored with list of relative rank for the current \texttt{\edtext}. That means, in the previous example, that, if the absolute line number of \texttt{\edtext} was 1, that critical notes (A) and (B) were not associated with \{1, 2, 2, 3\} but with \{(1, 1), (2, 1), (2, 1), (3, 1)\}. Such method to know the absolute line number associated to a \texttt{\sameword} is required because a \texttt{\edtext} can be overlap many lines, but \texttt{\sameword} can get it.
    * We get the value associated, when reading the auxiliary file, to the pair compose by the current marked word and the current absolute line number. Let this value be \texttt{n}.
  - If \texttt{n} > 1, that mean the current word appears more than once time in its line. In this case, we call \texttt{\showwordrank} with the word as first argument and \texttt{r} as second argument. If the word is called only once, we just print it.

After theory, implementation.
As the argument of \sameword can contain active character if we use inputenc with utf8 option instead of native UTF-8 engine, we store its detokenized content in a macro in order to allow dynamic name of macro with \csname.\footnote{See http://tex.stackexchange.com/q/244538/7712}

Because there is a bug with \detokenize and XƎTEX when using non BMP characters\footnote{http://sourceforge.net/p/xetex/bugs/108/}, we detokenize only for not XƎTEX engines. In any case, in XƎTEX, a \csname construction can contain UTF-8 characters without a problem, as UTF-8 characters are not managed with category code, but instead read directly as UTF-8 characters.

\begin{verbatim}
1078 \newcommand{\get@sw@txt}[1]{%
1079 \ifxetex%
1080 \xdef\sw@txt{#1}%
1081 \else%
1082 \expandafter\xdef\expandafter\sw@txt\expandafter{\detokenize{#1}}%
1083 \fi%
1084 }%
\end{verbatim}

\sameword The high level macro \sameword, used by the editor.

\begin{verbatim}
1085 \newcommandr{\sameword}[2][1,usedefault]{%
1086 \leavevmode%
1087 \get@sw@txt{#2}%
1088 % Now, the real code. First, increment the counter corresponding to the argument.
1089 \unless\ifledRcol%
1090 \csnumgdef{sw@\sw@txt}{\csuse{sw@\sw@txt}+1}%
1091 \fi%
1092 % Then, write its value to the numbered file.
1093 \protected@write\linenum@out{}{\string\@sw{\sw@txt}{\csuse{sw@\sw@txt}}{#1}}%
1094 % Do the same thing if we are in the right columns.
1095 \else%
1096 \csnumgdef{sw@\sw@txt@R}{\csuse{sw@\sw@txt@R}+1}%
1097 \protected@write\linenum@outR{}{\string\@sw{\sw@txt}{\csuse{sw@\sw@txt@R}}{#1}}%
1098 \fi%
1099 % And print the word.
1100 #2%
1101 }%
1102 \end{verbatim}

A flag set to true if a \@sw relative rank must be added to the list of ranks for a specific \edtext.

\begin{verbatim}
\if@addsw
1103 \newif\if@addsw%
\end{verbatim}

\@sw The command printed in the auxiliary files.

\begin{verbatim}
1104 \newcommand{\@sw}[3]{%
1105 \get@sw@txt{#1}%
1106 \unless\ifledRcol%
1107 \csnumgdef{sw@\sw@txt}{\csuse{sw@\sw@txt}+1}%
1108 \protected@write\linenum@out{}{\string\@sw{\sw@txt}{\csuse{sw@\sw@txt}}{#1}}%
1109 \fi%
1110 \end{verbatim}
First, define a counter which store the second argument as value for a each paired absolute line number/first argument

\csxdef{sw@\sw@txt @\the\absline@num @\the\section@num}{#2}%

If such argument was not defined for the preceding line, define it.
\numdef{\prev@line}{\the\absline@num-1}%
\ifsnumberdef{sw@\sw@txt @\prev@line @\the\section@num}{%}
\csnumgdef{sw@\sw@txt @\prev@line @\the\section@num}{#2-1}%
\}%

Then, calculate the position of the word in the line.
\numdef{\the@sw}{#2-\csuse{sw@\sw@txt @\prev@line @\the\section@num}}%

And do the same thing for the right side.
\else%
\csxdef{sw@\sw@txt @\the\absline@numR @\the\section@numR @R}{#2}%
\numdef{\prev@line}{\the\absline@numR-1}%
\ifsnumberdef{sw@\sw@txt @\prev@line @\the\section@numR @R}{%}
\csnumgdef{sw@\sw@txt @\prev@line @\the\section@numR @R}{#2-1}%
\}%
\numdef{\the@sw}{#2-\csuse{sw@\sw@txt @\prev@line @\the\section@numR @R}}%
\fi%

And now, add it to the list of \@sw for the current edtext, in all depth.
\@tempcnta=\@edtext@level
\@whilenum{\@tempcnta>0}\do{% 
\ifsnumberdef{sw@list@edtext@tmp@\the\@tempcnta}{%}
\letcs{\@addswfalse}{\notbool{lemmacommand@\the\@tempcnta}%}
{\@addswtrue}%
{\IfStrEq{#3}{inlemma}{\@addswtrue}{%}
{\def\do##1{%}
\ifnumequal{##1}{\the\@tempcnta}{\@addswtrue\listbreak}{}%}
docs\list(#3)%}
\if@addsw%
\letcs{\@tmp}{sw@list@edtext@tmp@\the\@tempcnta}%
\ifledRcol%
\xright@appenditem{{\the@sw}{\the\absline@numR}}\to\@tmp%
\else%
\xright@appenditem{{\the@sw}{\the\absline@num}}\to\@tmp%
\fi%
\cslet{sw@list@edtext@tmp@\the\@tempcnta}{\@tmp}%
\fi% }%
The command called when \sameword is called in an edtext.
\newcommandx{\sameword@inedtext}[2][1, usedefault]{% 
\get@sw@txt[#2] \unless\ifledRcol\% 
Just a precaution. 
\ifx\sw@list@inedtext\empty \def\the@sw{999} \def\this@absline{-99}\% \else \gl@p\sw@list@inedtext\to@tmp \edef\the@sw{\expandafter\@firstoftwo\@tmp} \edef\this@absline{\expandafter\@secondoftwo\@tmp}\% \fi \fi} 
First, calculate the number of occurrences of the word in the current line \sw@list@inedtext, which contains the reference for edtext. 
\ifcsdef{sw@\sw@txt @\this@absline @\the\section@num}{% 
\numdef{\prev@line}{\this@absline-1}\% 
\numdef{\sw@atthisline}{\csuse{sw@\sw@txt @\this@absline @\the\section@num}-\csuse{sw@\sw@txt @\prev@line @\the\section@num}}\% 
\} \% 
Finally, print the rank, but only if there is more than one occurrence of the word in the current line. 
\ifnumgreater{\sw@atthisline}{1}\% 
\{\showwordrank[#2]{\the@sw}\% 
\} \% 
And the same for right side. 
\else \fi\% 
\ifx\sw@list@inedtext\empty \def\the@sw{999} \def\this@absline{-99}\% \else \gl@p\sw@list@inedtext\to@tmp \edef\the@sw{\expandafter\@firstoftwo\@tmp} \edef\this@absline{\expandafter\@secondoftwo\@tmp}\% \fi \fi\% 
\ifcsdef{sw@\sw@txt @\this@absline @\the\section@numR @R}{% 
\numdef{\prev@line}{\this@absline-1}\% 
\numdef{\sw@atthisline}{\csuse{sw@\sw@txt @\this@absline @\the\section@numR @R}-\csuse{sw@\sw@txt @\prev@line @\the\section@numR @R}}\% 
\} \%
23 Paragraph decomposition and reassembly

In order to be able to count the lines of text and affix line numbers, we add an extra stage of processing for each paragraph. We send the paragraph into a box register, rather than straight onto the vertical list, and when the paragraph ends we slice the paragraph into its component lines; to each line we add any notes or line numbers, add a command to write to the line-list, and then at last send the line to the vertical list. This section contains all the code for this processing.

23.1 Boxes, counters, \pstart and \pend

\raw@text \ifnumberedpar@ \numberedpar@true \numberedpar@false \num@lines \one@line \par@line

Here are numbers and flags that are used internally in the course of the paragraph decomposition. When we first form the paragraph, it goes into a box register, \raw@text, instead of onto the current vertical list. The \ifnumberedpar@ flag will be true while a paragraph is being processed in that way. \num@lines will store the number of lines in the paragraph when it’s complete. When we chop it up into lines, each line in turn goes into the \one@line register, and \par@line will be the number of that line within the paragraph.

\pstart \AtEveryPstart \numberpstarttrue \numberpstartfalse \labelpstarttrue \labelpstartfalse \thepstart

\pstart starts the paragraph by clearing the \inserts@list list and other relevant variables, and then arranges for the subsequent text to go into the \raw@text box. \pstart needs to appear at the start of every paragraph that’s to be numbered; the \autopar command below may be used to insert these commands automatically.

Beware: everything that occurs between \pstart and \pend is happening within a group; definitions must be global if you want them to survive past the end of the paragraph.
\newcommand\AtEveryPstart\[1\]{\xdef\at@every@pstart{\noindent\unexpanded{#1}}}%
\xdef\at@every@pstart{}%
\newcounter{pstart}
\renewcommand\thepstart{{\bfseries\@arabic\c@pstart}. }
\newif\ifnumberpstart
\numberpstartfalse
\newif\iflabelpstart
\labelpstartfalse
\newcommandx*{\pstart}[1][1][1]{
\normal@pars
\ifstrempty{#1}{\at@every@pstart}{\noindent#1}%
\ifautopar
\autopar
\fi%
\ifluatex%
\edef\l@luatextextdir@L{\the\luatextextdir}%
\fi%
\if@nobreak
\let\@oldnobreak\@nobreaktrue%
\else%
\let\@oldnobreak\@nobreakfalse%
\fi%
\@nobreaktrue%
\ifnumbering \else%
\le@err@PstartNotNumbered%
\beginnumbering%
\fi%
\ifnumberedpar%
\le@err@PstartInPstart%
\pend%
\list@clear{\inserts@list}%
\global\let\next@insert=\empty%
\begingroup\normal@pars%
\global\advance \l@dnumpstartsL\@ne
\global\setbox\raw@text=\vbox\bgroup%
\ifautopar\else%
\ifnumberpstart%
\ifinstanza\else%
\ifsidestartnum\else%
\thepstart%
\fi%
\fi%
\fi%
\fi%
\fi%
\global\setbox\@numpstartsL\@ne
\global\setbox\raw@text=\vbox\bgroup%
\ifautopar\else%
\ifnumberpstart
\protected@edef\@currentlabel{\p@pstart\thepstart}
\section{Boxes, counters, \texttt{pstart} and \texttt{pend}}

\begin{verbatim}
\fi\l@dzeropenalties\
\endgraf\global\par@line=0%

We check if lineation is by pstart: in this case, we reset line number, but only in
the second line of the pstart, to prevent some trouble. We can’t reset line number
at the beginning of \texttt{pstart} \texttt{\setline} is parsed at the end of previous \texttt{pend}, and
so, we must do it at the end of first line of pstart.

\csnumdef{pstartline}{0}%
\loop\ifvbox\raw@text%
\csnumdef{pstartline}{\pstartline+1}%
\do@line%
\ifbypstart%
\ifnumequal{\pstartline}{1}{\setline{1}\resetprevline@}{}%
\fi%
\repeat%

Deal with any leftover notes, and then end the group that was begun in the
\texttt{pstart}.

\flush@notes%
\endgroup%
\ignorespaces%
\ifnumberpstart%
\pstartnumtrue%
\fi%
\@oldnobreak%
\addtocounter{pstart}{1}%
\normalpars%
\ifstrempty{#1}{\at@every@pend}{\noindent#1}%
\ifautopar%
\autopar%
\end{verbatim}

\pend \pend must be used to end a numbered paragraph.

\begin{verbatim}
\newcommandx*{\pend}[1][1]{\ifnumbering \else%
\led@err@PendNotNumbered%
\fi%
\global\l@dskipversenumberfalse%
\ifnumberedpar@ \else%
\led@err@PendNoPstart%
\fi%
\fi%

We set all the usual interline penalties to zero and then immediately call \texttt{endgraf}
to end the paragraph; this ensures that there’ll be no large interline penalties to
prevent us from slicing the paragraph into pieces. These penalties revert to the
values that you set when the group for the \texttt{vbox} ends. Then we call \texttt{do@line}
to slice a line off the top of the paragraph, add a line number and footnotes, and
restore it to the page; we keep doing this until there aren’t any more lines left.

\l@dzeropenalties%
\endgraf\global\num@lines=\prevgraf\egroup%
\global\par@line=0%
\end{verbatim}
\AtEveryPend
\@everypend\{
\newcommand{\AtEveryPend}{\noindent\unexpanded{#1}}%
\newcommand{\AtEveryPend}{%}
\xdef\at\everypend{%}
\l@dzeropenalties A macro to zero penalties for \pend or \pstart.
\newcommand*{\l@dzeropenalties}{% 
\brokenpenalty \z@ \clubpenalty \z@ \displaywidowpenalty \z@ \interlinepenalty \z@ \predisplaypenalty \z@ \postdisplaypenalty \z@ \widowpenalty \z@}{
\autopar In most cases it’s only an annoyance to have to label the paragraphs to be numbered with \pstart and \pend. \autopar will do that automatically, allowing you to start a paragraph with its first word and no other preliminaries, and to end it with a blank line or a \par command. The command should be issued within a group, after \beginnumbering has been used to start the numbering; all paragraphs within the group will be affected.

A few situations can cause problems. One is a paragraph that begins with a begin-group character or command: \pstart will not get invoked until after such a group beginning is processed; as a result the character that ends the group will be mistaken for the end of the \texttt{vbox} that \pstart creates, and the rest of the paragraph will not be numbered. Such paragraphs need to be started explicitly using \texttt{indent}, \texttt{noindent}, or \texttt{leavevmode}—or \pstart, since you can still include your own \pstart and \pend commands even with \autopar on.

Prematurely ending the group within which \autopar is in effect will cause a similar problem. You must either leave a blank line or use \par to end the last paragraph before you end the group.

The functioning of this macro is more tricky than the usual \texttt{everypar}: we don’t want anything to go onto the vertical list at all, so we have to end the paragraph, erase any evidence that it ever existed, and start it again using \pstart. We remove the paragraph-indentation box using \texttt{lastbox} and save the width, and then skip backwards over the \parskip that’s been added for this paragraph. Then we start again with \pstart, restoring the indentation that we saved, and locally change \par so that it’ll do our \pend for us.
\newif\ifautopar
\autoparfalse
\newcommand*{\autopar}{% 
\ifledRcol % 
\ifnumberingR \else
\le@err@AutoparNotNumbered
\beginnumberingR
\newif\ifautopar
\autoparfalse
\newcommand*{\autopar}{% 
\ifledRcol % 
\ifnumberingR \else
\le@err@AutoparNotNumbered
\beginnumberingR
We also define a macro which we can rely on to turn off the \autopar definitions at various important places, if they are in force. We'll want to do this within a footnotes, for example.

\newcommand*{\normal@pars}{\everypar{}\let\par\endgraf}  

\ifautopar@pause We define a boolean test switched to true at the beginning of the \pausenumbering command if the autopar is enabled. This boolean will be tested at the beginning of \resumenumbering to continue the autopar if needed.

\newif{\ifautopar@pause}

23.2 Processing one line

The \do@line macro is called by \pend to do all the processing for a single line of text.

\newcommand*{\l@dunhbox@line}[1]{\unhbox #1}  
\newcommand*{\do@line}{\vbadness=10000\splittopskip=\z@\l@demptydata\global\setbox\one@line=\vsplit\raw@text to\baselineskip}  
\unvbox\one@line\global\setbox\one@line=\lastbox\getline@num\IfStrEq{\led@pb@setting}{before}{}\else{}\fi\inserthangingsymboltrue\check@pb@in@verse\ifl@dhidenumber%\global\l@dhidenumberfalse%\f@x@l@cks%\else%
Paragraph decomposition and reassembly

Depending whether a sectioning command is called at this pstart or not we print
sectioning command or normal line,

\ifinlist{\the\l@dnumpstarts\l}{\eled@sections@@}\
{\print@eledsection}\
{\print@line}\
\IfStrEq{\led@pb@setting}{after}{\led@check@pb\led@check@nopb}{}\fi\%}
\print@line

\print@line is for normal line, i.e. line without sectioning command.

\def\print@line{
  Insert the pstart number in side, if we are in the first line of a pstart.
  \affixpstart@num\%

  The line will be boxed, to have the good width.
  \hb@xt@ \linewidth{...

  User hook.
  \do@insidelinehook%

  Left line number
  \l@dld@ta

  Restore marginal and footnotes.
  \add@inserts\affixside@note\%

  Print left notes.
  \l@dlsn@te

  Boxes the line, writes information about new line in the numbered file.
  \ledllfill\hb@xt@ \wd\one@line{\new@line%

  If we use Lua\TeX \ then restore the direction.
  \ifluatex%
  \luatextextdir\l@luatextextdir@L\%
  \fi%

  Insert, if needed, the hanging symbol.
  \inserthangingsymbol \%

  And so, print the line.
  \l@dunhbox@line\one@line}%

  Right line number
  \ledrlfill\l@drd@ta

  Print right notes.
  \l@drsn@te

  And reinsert penalties (for page breaking)...
  \add@penalties\%}

\print@eledsection \print@eledsection to print sectioning command with line number. It sets the correct spacing, depending whether a sectioning command was called at previous \pstart, calls the sectioning command, prints the normal line outside of the paper, to be able to have critical footnotes. Because of how this prints, a vertical spacing correction is added.

1357 \def\print@eledsection{%
1358 \add@inserts\affixside@note%
1359 \numdef{\temp0}{l@dnumpstartsL-1}%
1360 \xifinlist{\temp0}{eled@sections@@}{\@nobreaktrue}{\@nobreakfalse}%
1361 \@eled@sectioningtrue%
1362 \csuse{eled@sectioning@the\l@dnumpstartsL}%
1363 \@eled@sectioningfalse%
1364 \global\csundef{eled@sectioning@the\l@dnumpstartsL}%
1365 \if@RTL%
1366 \hspace{-3\paperwidth}%
1367 \{\hbox{\l@dunhbox@line{one\line}} \new@line}%
1368 \else%
1369 \hspace{3\paperwidth}%
1370 \{\new@line \hbox{\l@dunhbox@line{one\line}}}%
1371 \fi%
1372 \vskip-\baselineskip%
1373 }

These hight level commands just redefine the low level commands. They have to be used be user, without \makeatletter.

1374 \newcommand*{\dolinehook}{[1]{\gdef\do@linehook{#1}}%
1375 \newcommand*{\doinsidelinehook}{[1]{\gdef\do@insidelinehook{#1}}%
1376
\dolinehook Two hooks into \do@line. The first is called at the beginning of \do@line, the second is called in the line box. The second can, for example, have a \markboth command inside, the first can’t.

1377 \newcommand*{\do@linehook}{}
1378 \newcommand*{\do@insidelinehook}{}

\do@linehook Nulls the \...d@ta, which may later hold line numbers. Similarly for \do@csnotetext, \do@csnotetext0, \do@csnotetext0 for the texts of the sidenotes, left and right notes.

1379 \newcommand*{\do@emptyd@ta}{% 
1380 \gdef\l@dld@ta{}
1381 \gdef\l@drd@ta{}
1382 \gdef\l@dcsnotetext@l{} 
1383 \gdef\l@dcsnotetext@r{} 
1384 \gdef\l@dcsnotetext{} 

\do@insidelinehook Zero width boxes of the left and right side notes, together with their kerns.

1385 \newcommand{\l@dlsn@te}{
1386 \newcommand{\l@dlsn@te}{

These macros are called at the left ($\texttt{\textbackslash ledllfill}$) and the right ($\texttt{\textbackslash ledrrfill}$) of each numbered line. The initial definitions correspond to the original code for \do@line.

\newcommand\ledllfill{\hfil}
\newcommand\ledrrfill{}

23.3 Line and page number computation

\getline@num The \getline@num macro determines the page and line numbers for the line we’re about to send to the vertical list.

\newcommand*{\getline@num}{\global\advance\absline@num 1\do@actions\do@ballast\ifnumberline\ifnum\@lock<\tw@\global\advance\subline@num 1\else\ifnum\@lock<\tw@\global\advance\line@num 1\global\subline@num 0\else\fi\fi\fi;}

\do@ballast The real work in the macro above is done in \do@actions, but before we plunge into that, let’s get \do@ballast out of the way. This macro looks to see if there is an action to be performed on the next line, and if it is going to be a page break action, \do@ballast decreases the count \ballast@count counter by the amount of \ballast. This means, in practice, that when \add@penalties assigns penalties at this point, $\LaTeX$ will be given extra encouragement to break the page here (see \ref{sec:pagebreak} p.\pageref{sec:pagebreak}).

\ballast@count First we set up the required counters; they are initially set to zero, and will remain \ballast@count so unless you say \setcounter{ballast}{⟨some figure⟩} in your document.

\newcounter{ballast}
\setcounter{ballast}{0}
And here is \do@ballast itself. It advances \absline@num within the protection of a group to make its check for what happens on the next line.

\newcommand*{\do@ballast}{\global\ballast@count \z@}
\begingroup
\advance\absline@num \@ne
\ifnum\next@actionline=\absline@num
\ifnum\next@action>-1001\relax
\fi
\fi
\endgroup}

The \do@actions macro looks at the list of actions to take at particular absolute line numbers, and does everything that’s specified for the current line.

It may call itself recursively, and to do this efficiently (using \TeX’s optimization for tail recursion), we define a control-sequence called \do@actions@next that is always the last thing that \do@actions does. If there could be more actions to process for this line, \do@actions@next is set equal to \do@actions; otherwise it’s just \relax.

\newcommand*{\do@actions}{%
\global\let\do@actions@next=\relax
\ifnum\absline@num<\next@actionline\else
\fi}

First, page number changes, which will generally be the most common actions.

If we’re restarting lineation on each page, this is where it happens.

\ifnum\next@actionline>-1001
\global\page@num=\next@action
\ifbypage@
\global\line@num=\z@ \global\subline@num=\z@
\resetprevline@
\fi
\fi

Next, we handle commands that change the line-number values. (We subtract 5001 rather than 5000 here because the line number is going to be incremented automatically in \getline@num.)

\else
\ifnum\next@actionline<-4999
\@oldt@mpc@nta=-\next@action
\advance\@oldt@mpc@nta by -5001
\ifs@blines@\global\subline@num=\@oldt@mpc@nta\else
\global\line@num=\@oldt@mpc@nta\fi
\fi

It’s one of the fixed codes. We rescale the value in \@oldt@mpc@nta so that we can use a case statement.

\else
\@oldt@mpc@nta=-\next@action
\advance\@oldt@mpc@nta by -1000
\fi
Now we get information about the next action off the list, and then set \do@actions@next so that we’ll call ourself recursively: the next action might also be for this line.

There’s no warning if we find \actionlines@list empty, since that will always happen near the end of the section.

Make the recursive call, if necessary.

\do@actions@fixedcode

This macro handles the fixed codes for \do@actions. It is one big case statement.
\affixline@num \affixline@num originally took a single argument, a series of commands for printing the line just split off by \do@line: it put that line back on the vertical list, and added a line number if necessary. It now just puts a left line number into \l@dld@ta or a right line number into \l@drd@ta if required.

To determine whether we need to affix a line number to this line, we compute the following:

\[ n = \text{int}\left(\frac{\text{linenum} - \text{firstlinenum}}{\text{linenumincrement}}\right) \]
\[ m = \text{firstlinenum} + (n \times \text{linenumincrement}) \]

(where int truncates a real number to an integer). \( m \) will be equal to \( \text{linenum} \) only if we’re to paste a number on here. However, the formula breaks down for the first line to number (and any before that), so we check that case separately: if \( \text{line@num} \leq \text{firstlinenum} \), we compare the two directly instead of making these calculations.

We compute, in the scratch counter \@l@dtempcnta, the number of the next line that should be printed with a number (\( m \) in the above discussion), and move the current line number into the counter \@l@dtempcntb for comparison.

First, the case when we’re within a sub-line range.

\newcommand*{\affixline@num}{%}

No number is attached if \ifd@skipnumber is TRUE (and then it is set to its normal FALSE value). No number is attached if \ifnumberline is FALSE (the normal value is TRUE).

\ifledgroupnotesL@\else
\ifnumberline
\ifd@skipnumber
\global\d@skipnumberfalse
\else
\ifs@ublines@\else
\@dtempcntb=\subline@num
\ifnum\subline@num>c@firstsublinenum
\@dtempcnta=\subline@num
\advance\@dtempcnta by-c@firstsublinenum
\divide\@dtempcnta by@c@sublinenumincrement
\multiply\@dtempcnta by@c@sublinenumincrement
\advance\@dtempcnta by@c@firstsublinenum
\else

That takes care of computing the values for comparison, but if line number locking is in effect we have to make a further check. If this check fails, then we disable the line-number display by setting the counters to arbitrary but unequal values.

Now the line number case, which works the same way.

Check on the \linenumberlist If it's \empty use the standard algorithm.

The \linenumberlist wasn't \empty, so here's Wayne's numbering mechanism. This takes place in TeX's mouth.

A locking check for lines, just like the version for sub-line numbers above.

The following tests are true if we need to print a line number.
produces a number that's even for left-margin numbers and odd for right-margin numbers.

For \TeX{} we have to consider two column documents as well. In this case I think we need to put the numbers at the outside of the column — the left of the first column and the right of the second. Do the two column stuff before going on with the original code.

\@ldldota\ A left line number is stored in \@ldldota\ and a right one in \@drdota.\n\@drdota
1534     \if@twocolumn
1535         \if@firstcolumn
1536             \gdef\@ldldota{\llap{{\leftlinenum}}}%
1537         \else
1538             \gdef\@drdota{\rlap{{\rightlinenum}}}%
1539         \fi
1540     \else
1541     \fi
1542
Continuing the original code ...
1543     \@l@dtempcntb=\line@margin
1544     \ifnum\@l@dtempcntb>\@ne
1545         \advance\@l@dtempcntb \page@num
1546     \fi
1547
Now print the line (#1) with its page number.
1548     \ifodd\@l@dtempcntb
1549         \gdef\@drdota{\rlap{{\rightlinenum}}}%
1550     \else
1551         \gdef\@ldldota{\llap{{\leftlinenum}}}%
1552     \fi
1553
Now fix the lock counters, if necessary. A value of 1 is advanced to 2; 3 advances to 0; other values are unchanged.
1554     \f@x@l@cks
1555     \fi
1556     \fi
1557 }
1558
\ch@cksub@l@ck These macros handle line number locking for \affixline@num. \ch@ck@l@ck checks subline locking. If it fails, then we disable the line-number display by setting the counters to arbitrary but unequal values.
1559 \newcommand*{\ch@cksub@l@ck}{%\n1560     \ifcase\sub@lock
1561         \or
1562             \ifnum\sublock@disp=\@ne
1563                 \@l@dtempcntb=\z@ \@l@dtempcnta=\@ne
Similarly for line numbers.

\newcommand*{\ch@ck@l@ck}{%
\ifcase\@lock
\or
\ifnum\lock@disp=\@ne
\@l@dtempcntb=\z@ \@l@dtempcnta=\@ne
\fi
\or
\ifnum\lock@disp=\tw@
\else
\fi
\fi
\fi}

Fix the lock counters. A value of 1 is advanced to 2; 3 advances to 0; other values are unchanged.

\newcommand*{\f@x@l@cks}{%
\ifcase\@lock
\or
\global\@lock=\tw@
\or \or
\global\@lock=\z@
\fi
\fi
\fi
\fi}

\pageparbreak

Because of TeX’s asynchronous page breaking mechanism we can never be sure just where it will make a break and, naturally, it has already decided exactly how it will typeset any remainder of a paragraph that crosses the break. This is disconcerting when trying to number lines by the page or put line numbers in different margins. This macro tries to force an invisible paragraph break and a
24.1 Pstart number printing in side

In side, the printing of pstart number is running like the printing of line number. There is only some differences:

- The pstarts counter is upgrade in the \pend command. Consequently, the \affixpstart@num command has not to upgrade it, unlike the \affixline@num which upgrades the lines counter.

- To print the pstart number only at the beginning of a pstart, and not in every line, a boolean test is made. The \pstartnum boolean is set to TRUE at every \pend. It’s tried in the \leftpstartnum and \rightstartnum commands. After the try, it is set to FALSE.
24.2 Add insertions to the vertical list

\inserts@list \inserts@list is the list macro that contains the inserts that we save up for one paragraph.
\list@create{\inserts@list}

\add@inserts \add@inserts is the penultimate macro used by \do@line; it takes insertions saved in a list macro and sends them onto the vertical list.

It may call itself recursively, and to do this efficiently (using \TeX’s optimization for tail recursion), we define a control-sequence called \add@inserts@next that is always the last thing that \add@inserts does. If there could be more inserts to process for this line, \add@inserts@next is set equal to \add@inserts; otherwise it’s just \relax.
\newcommand*{\add@inserts}{%
\global\let\add@inserts@next=\relax
If \inserts@list is empty, there aren’t any more notes or insertions for this paragraph, and we needn’t waste our time.
\ifx\inserts@list\empty \else
The \next@insert macro records the number of the line that receives the next footnote or other insert; it’s empty when we start out, and just after we’ve affixed a note or insert.
\ifnum\next@insert=\absline@num
If the next insert’s for this line, tack it on (and then erase the contents of the insert macro, as it could be quite large). In that case, we also set \add@inserts@next so that we’ll call ourselves recursively: there might be another insert for this same line.
\ifnum\next@insert=\absline@num
24.3 Penalties

\addpenalties is the last macro used by \do@line. It adds up the club, widow, and interline penalties, and puts a single penalty of the appropriate size back into the paragraph; these penalties get removed by the \vsplit operation. \displaywidowpenalty and \brokenpenalty are not restored, since we have no easy way to find out where we should insert them.

In this code, \num@lines is the number of lines in the whole paragraph, and \par@line is the line we’re working on at the moment. The count \@@dtempcnta is used to calculate and accumulate the penalty; it is initially set to the value of \ballast@count, which has been worked out in \do@ballast above (23.3 p. 111). Finally, the penalty is checked to see that it doesn’t go below \penalty -10000.

\newcommand*{\addpenalties}{\@l@dtempcnta=\ballast@count
\ifnum\num@lines>\@ne
\global\advance\par@line \@ne
\ifnum\par@line=\@ne
\advance\@l@dtempcnta \clubpenalty
\fi
\@l@dtempcntb=\par@line \advance\@l@dtempcntb \@ne
\ifnum\@l@dtempcntb=\num@lines
\advance\@l@dtempcnta \widowpenalty
\fi
\ifnum\par@line<\num@lines
\advance\@l@dtempcnta \interlinepenalty
\fi
\fi
\ifnum\@l@dtempcnta=\z@\relax
\else
\ifnum\@l@dtempcnta<-10000 \penalty\@l@dtempcnta
\else
\penalty -10000
\fi
\fi
\fi}
24.4 Printing leftover notes

The \flush@notes macro is called after the entire paragraph has been sliced up and sent on to the vertical list. If the number of notes to this paragraph has increased since the last run of \TeX{}, then there can be leftover notes that haven’t yet been printed. An appropriate error message will be printed elsewhere; but it’s best to go ahead and print these notes somewhere, even if it’s not in quite the right place. What we do is dump them all out here, so that they should be printed on the same page as the last line of the paragraph. We can hope that’s not too far from the proper location, to which they’ll move on the next run.

\begin{verbatim}
1691 \newcommand*{\flush@notes}{%
1692 \@xloop
1693 \if\inserts@list\empty \else
1694 \global\inserts@list\to\@insert
1695 \@insert
1696 \global\let\@insert=\undefined
1697 \repeat}
\end{verbatim}

\flush@notes \@xloop is a variant of the Plain \TeX{} \loop macro, useful when it’s hard to construct a positive test using the \TeX{} \if commands—as in \flush@notes above. One says \@xloop \ldots \if \ldots \else \ldots \repeat, and the action following \else is repeated as long as the \if test fails. (This macro will work wherever the Plain \TeX{} \loop is used, too, so we could just call it \loop; but it seems preferable not to change the definitions of any of the standard macros.)

This variant of \loop was introduced by Alois Kabelschacht in \textit{TUGboat} 8 (1987), pp. 184–5.

25 Critical footnotes

The footnote macros are adapted from those in Plain \TeX{}, but they differ in these respects: the outer-level commands must add other commands to a list macro rather than doing insertions immediately; there are five separate levels of the footnotes, not just one; and there are options to reformat footnotes into paragraphs or into multiple columns.

25.1 Fonts

Before getting into the details of formatting the notes, we set up some font macros. It is the notes that present the greatest challenge for our font-handling mechanism, because we need to be able to take fragments of our main text and print them in different forms: it is common to reduce the size, for example, without otherwise changing the fonts used.
\select@lemmafont is provided to set the right font for the lemma in a note. This macro extracts the font specifier from the line and page number cluster, and issues the associated font-changing command, so that the lemma is printed in its original font.

\begin{verbatim}
def\select@lemmafont#1|#2|#3|#4|#5|#6|#7|{
\select@@lemmafont#7|}
def\select@@lemmafont#1/#2/#3/#4|%
{
\fontencoding{#1}\fontfamily{#2}\fontseries{#3}\fontshape{#4}|%
\selectfont}
\end{verbatim}

25.2 Outer-level footnote commands

The \footnoteoptions@ change the value of on options of Xfootnote, to switch between true and false.

\begin{verbatim}
def\xnewcommandx*{\footnoteoptions@}[3][1=L,usedefault]{%
def\do##1{%
  \ifstrequal{#1}{L}{% In Leftside
    \xright@appenditem{{\global
      \noexpand\settoggle{##1@}{#3}}}\to\inserts@list% Switch toogle, in all case
    \global\advance\insert@count \@ne% Increment the left insert counter.
  }%
  {%
    \xright@appenditem{{\global
      \noexpand\settoggle{##1@}{#3}}}\to\inserts@listR% Switch toogle, in all case
    \global\advance\insert@countR \@ne% Increment the right insert counter insert.
  }%
}%
\notblank{#2}{\docsvlist{#2}}{}% Parsing all options
}
\end{verbatim}

\footnotelang@lua is called to remember the information about the language of a lemma when LuaLaTeX is used.

\begin{verbatim}
def\xnewcommandx*{\footnotelang@lua}[1][1=L,usedefault]{%
  \ifstrequal{#1}{L}{% In Leftside
    \xright@appenditem{{\csxdef{footnote@luatextextdir}{\the\luatextextdir}}}\to\inserts@list% Know the dir of lemma
    \global\advance\insert@count \@ne%
    \xright@appenditem{{\csxdef{footnote@luatexpardir}{\the\luatexpardir}}}\to\inserts@list% Know the dir of lemma
    \global\advance\insert@count \@ne%
  }%
  {%
    \xright@appenditem{{\csxdef{footnote@luatextextdir}{\the\luatextextdir}}}\to\inserts@listR% Know the dir of lemma
    \global\advance\insert@countR \@ne%
    \xright@appenditem{{\csxdef{footnote@luatexpardir}{\the\luatexpardir}}}\to\inserts@listR% Know the dir of lemma
    \global\advance\insert@countR \@ne%
  }%
}
\end{verbatim}

\footnotelang@poly is called to remember the information about the language of a lemma when Polyglossia is used.

\begin{verbatim}
def\xnewcommandx*{\footnotelang@poly}[1][1=L,usedefault]{%
  \ifstrequal{#1}{L}{% In Leftside
    \xright@appenditem{{\csxdef{footnote@luatextextdir}{\the\luatextextdir}}}\to\inserts@listR% Know the dir of lemma
    \global\advance\insert@countR \@ne%
    \xright@appenditem{{\csxdef{footnote@luatexpardir}{\the\luatexpardir}}}\to\inserts@listR% Know the dir of lemma
    \global\advance\insert@countR \@ne%
  }%
  {%
    \xright@appenditem{{\csxdef{footnote@luatextextdir}{\the\luatextextdir}}}\to\inserts@list% Know the dir of lemma
    \global\advance\insert@count \@ne%
    \xright@appenditem{{\csxdef{footnote@luatexpardir}{\the\luatexpardir}}}\to\inserts@list% Know the dir of lemma
    \global\advance\insert@count \@ne%
  }%
}
\end{verbatim}
25.3 Normal footnote formatting

The processing of each note is done by four principal macros: the \vfootnote macro takes the text of the footnote and does the \insert; it calls on the \footfmt macro to select the right fonts, print the line number and lemma, and do any other formatting needed for that individual note. Within the output routine, the two other macros, \footstart and \footgroup, are called; the first prints extra vertical space and a footnote rule, if desired; the second does any reformatting of the whole set of the footnotes in this series for this page—such as paragraphing or division into columns—and then sends them to the page.

These four macros, and the other macros and parameters shown here, are distinguished by the ‘series letter’ that indicates which set of the footnotes we’re dealing with—A, B, C, D, or E. The series letter always precedes the string foot in macro and parameter names. Hence, for the A series, the four macros are called \vAfootnote, \Afootfmt, \Afootstart, and \Afootgroup.

We now begin a series of commands that do ‘normal’ footnote formatting: a format much like that implemented in Plain \TeX{}, in which each footnote is a separate paragraph.

\normalvfootnote takes the series letter as #1, and the entire text of the footnote is #2. It does the \insert for this note, calling on the \footfmt macro for this note series to format the text of the note.
25.3 Normal footnote formatting

\csuse{bhookXnote@#1}
\csuse{Xnotefontsize@#1}
\footnotesplitsskips
\ifl@dpairing\ifl@dpaging\else\%
\setnoteswidthliketwocolumns@{#1}\%
\fi\fi%
\setnotespositionliketwocolumns@{#1}%
\spaceskip=\z@skip \xspaceskip=\z@skip
\csname #1footfmt\endcsname #2[#1]\egroup}

\footnotesplitsskips Some setup code that is common for a variety of the footnotes. The setup is for:

- \interlinepenalty.
- \splittopskip (skip before last part of notes that flow from one page to another).
- \splitmaxdepth.
- \floatingpenalty, that is penalty values being added when a long note flows from one page to another. Here, we let it to 0 when we are processing parallel pages in \texttt{eledpar}, in order to allow notes to flow from left to right pages and \textit{vice-versa}. Otherwise, we let it to @\texttt{OMM}, which is the standard \LaTeX \texttt{floatingpenalty}.

\newcommand*{% \footnotesplitsskips}{% \interlinepenalty=% \interfootnotelinepenalty \unless\ifl@dprintingpages% \floatingpenalty=\@MM% \fi% \splittopskip=\ht\strutbox \splitmaxdepth=\dp\strutbox \leftskip=\z@skip \rightskip=\z@skip}
\mpnormalvfootnote And a somewhat different version for minipages.
\notbool{parapparatus@}{\newcommand}{\newcommand}{\mpnormalvfootnote}{2}{% \global\setbox\@nameuse{mp#1footins}\vbox{% \unvbox\@nameuse{mp#1footins}\csuse{bhookXnote@#1}\csuse{Xnotefontsize@#1}\hsize\columnwidth \@parboxrestore \color@begingroup \csname #1footfmt\endcsname #2[#1]\color@endgroup}}

\setnormalparstuff@common \setnormalparstuff \Xsetnormalparstuff \pedsetnormalparstuff X \normalfootfmt is a ‘normal’ macro to take the footnote line and page number information (see 21.3 p. 72), and the desired text, and output what’s to be printed. Argument #1 contains the line and page number information and lemma font
\normalfootfmt
specifier; \#2 is the lemma; \#3 is the note’s text. This version is very rudimentary—it uses \texttt{\texttt{printlines}} to print just the range of line numbers, followed by a square bracket, the lemma, and the note text; it’s intended to be copied and modified as necessary.

\texttt{\par} should always be redefined to \texttt{\texttt{endgraf}} within the format macro (this is what \texttt{\texttt{normal@pars}} does), to override tricky material in the main text to get the lines numbered automatically (as set up by \texttt{\texttt{autopar}}, for example).

1788 \newcommand*{\ledsetnormalparstuff}{%  
1789   \ledwar@\ledsetnormalparstuffDeprecated%  
1790   \ifluatex%  
1791     \luatextextdir\footnote@luatextextdir%  
1792     \luatexpardir\footnote@luatexpardir%  
1793     \fi%  
1794   \csuse{\csuse{footnote@dir}}%  
1795   \normal@pars%  
1796   \noindent \parfillskip \z@ \@plus 1fil)%  
1797 \newcommand*{\ledsetnormalparstuff@common}{%  
1798   \ifluatex%  
1799     \luatextextdir\footnote@luatextextdir%  
1800     \luatexpardir\footnote@luatexpardir%  
1801     \fi%  
1802   \csuse{\csuse{footnote@dir}}%  
1803   \normal@pars%  
1804   \parfillskip \z@ \@plus 1fil}%  
1805 \newcommand*{\Xledsetnormalparstuff}{[1]{%  
1806   \ledsetnormalparstuff@common%  
1807   \nottoggle{Xparindent@#1}{\noindent}{}%  
1808 }%  
1809 \newcommand*{\ledsetnormalparstuffX}{[1]{%  
1810   \ledsetnormalparstuff@common%  
1811   \nottoggle{parindentX@#1}{\noindent}{}%  
1812 }%  
1813 \notbool{parapparatus@}{\newcommandx*}{\newcommand}{\normalfootfmt}[4][4=Z]{%  
1814   \Xledsetnormalparstuff{#4}  
1815   \hangindent=\csuse{Xhangindent@#4}  
1816   \strut\{\printlinefootnote{#1}{#4}\}}%  
1817 }%  
1818 }%  
1819 \endashchar  
1820 \fullstop  
1821 \rbracket  

The fonts that are used for printing notes might not have the character mapping we expect: for example, the Computer Modern font that contains old-style numerals
does not contain an en-dash or square brackets, and its period and comma are in odd locations. To allow use of the standard footnote macros with such fonts, we use the following macros for certain characters.

The `\endashchar` macro is simply an en-dash from the normal font and is immune to changes in the surrounding font. The same goes for the full stop. These two are used in `\printlines`. The right bracket macro is the same again; it crops up in `\normalfootfmt` and the other footnote macros for controlling the format of the footnotes.

With polyglossia, each critical note has a `\footnote@lang` which shows the language of the lemma, and which can be used to switch the bracket from right to left.

\begin{verbatim}
def\endashchar{\textnormal{--}}
\newcommand*{\fullstop}{\textnormal{.}}
\newcommand*{\rbracket}{\textnormal{\csuse{text\csuse{footnote@lang}}}\ifluatex\ifdefstring{\footnote@luatextextdir}{TRT}{\thinspace[}{\thinspace]}\else\thinspace]\fi}%
\end{verbatim}

The `\printpstart` macro prints the pstart number for a note.

\begin{verbatim}
\newcommand{\printpstart}{0}{% \ifboolexpr{bool{l@dpairing} or bool{l@dprintingpages} or bool{l@dprintingcolumns}}{\ifledRcol\thepstartR\else\thepstartL\fi}{\thepstart}}%
\end{verbatim}

The `\printlines` macro prints the line numbers for a note—which, in the general case, is a rather complicated task. The seven parameters of the argument are the line numbers as stored in `\l@d@nums`, in the form described on 21.3 p. 72; the starting page, line, and sub-line numbers, followed by the ending page, line, and sub-line numbers, and then the font specifier for the lemma.

The original EDMAC code used several counters at this point, saying:

To simplify the logic, we use a lot of counters to tell us which numbers need to get printed (using 1 for yes, 0 for no, so that `\ifodd` tests for ‘yes’). The counter assignments are:

- `\@pnum` for page numbers;
• \@ssub for starting sub-line;
• \@elin for ending line;
• \@esl for ending sub-line; and
• \@dash for the dash between the starting and ending groups.

There’s no counter for the line number because it’s always printed.

\LaTeX{} tends to use a lot of counters and packages should try and minimise the number of new ones they create. In line with this Peter Wilson has reverted to traditional booleans.

Maïeul Rouquette has added \ifl@d@twolines and \ifl@d@morethantwolines to print a symbol which stands for “and subsequent” when there are two, three or more lines.

\setistwofollowinglines
The \ifistwofollowinglines boolean, used by the \twolines and related tools, is set to true by \setistwofollowinglines. This command takes the following arguments:
• #1 First page number.
• #2 First line number.
• #3 Last page number.
• #4 Last line number.

If #3 - #2 = 1, then that means the two lines are subsequent, and consequently \texttt{\ifistwofollowinglines} is set to true. However, if we use lineation by page, two given lines can be subsequent if:

• The first line number is equal to the last line number of the first page.
• The last line number is equal to 1.
• #3 - #1 is equal to 1.

\begin{verbatim}
\newif\ifistwofollowinglines
\newcommand{\setistwofollowinglines}[4]{%
  \ifcsdef{lastlinenumberon@#1}{% \numdef{\tmp}{\csuse{lastlinenumberon@#1}}}{\numdef{\tmp}{0}}%
  \istwofollowinglines@false
  \ifnumequal{#4-#2}{1}{\istwofollowinglines@true}{%
    \ifbypage@%
    \ifnumequal{#3-#1}{1}{%
      \ifnumequal{#2}{\tmp}{\ifnumequal{#4}{1}{\istwofollowinglines@true}{}}{}}%
    {}%
  }%
  \fi%
}%
\setprintlines
We print the page numbers only if: 1) we’re doing the lineation by page, and 2) the ending page number is different from the starting page number.

Just a reminder of the arguments:
\texttt{\setprintlines start-page | line | subline | end-page | line | subline | font}

The macro \texttt{\setprintlines} does the work of deciding what numbers should be printed. Its arguments are the same as the first 6 of \texttt{\printlines}.
\end{verbatim}
We print the ending line number if: (1) we’re printing the ending page number, or (2) it’s different from the starting line number.

We print the starting sub-line if it’s nonzero.

We print the ending sub-line if it’s nonzero and: (1) it’s different from the starting sub-line number, or (2) the ending line number is being printed.

However, if the \twolines is set for the current series, we don’t print the last line number.
Now we’re ready to print it all. If the lineation is by pstart, we print the pstart.

One subtlety left here is when to print a period between numbers. But the only instance in which this is tricky is for the ending sub-line number: it could come after the starting sub-line number (in which case we want only the dash) or after an ending line number (in which case we need to insert a period). So, first, print this start line number.

Then print the dash + end line number, or the range symbol.
\normalfootstart

\normalfootstart is a standard footnote-starting macro, called in the output routine whenever there are footnotes of this series to be printed: it skips a bit and then draws a rule.

Any \footstart macro must put onto the page something that takes up space exactly equal to the \skip\footins value for the associated series of notes. \TeX makes page computations based on that \skip value, and the output pages will suffer from spacing problems if what you add takes up a different amount of space.

But if the skip \preXnotes is greater than 0 pt, it's used instead of \skip\footins for the first printed series.

The \leftskip and \rightskip values are both zeroed here. Similarly, these skips are cancelled in the \vfootnote macros for the various types of notes. Strictly speaking, this is necessary only if you are using paragraphed footnotes, but we have put it here and in the other \vfootnote macros too so that the behavior of eledmac in this respect is general across all footnote types (you can change this). What this means is that any \leftskip and \rightskip you specify applies to the main text, but not the footnotes. The footnotes continue to be of width \hsize.

\newcommand*{\normalfootstart}{%
Standard footnote definitions

\normalfootnoterule \normalfootnoterule is a standard footnote-rule macro, for use by a footstart macro: just the same as the Plain \TeX\ footnote rule.

\let\normalfootnoterule=\footnoterule

\normalfootgroup \normalfootgroup is a standard footnote-grouping macro: it sends the contents of the footnote-insert box to the output page without alteration.

\newcommand*{\normalfootgroup}[1]{% 
\csuse{Xnotefontsize@#1}\noindent\csuse{txtbeforeXnotes@#1}}% 
\unvbox\csname #1footins\endcsname% 
\hsize=\old@hsize%

A somewhat different version for minipages.

\newcommand*{\mpnormalfootgroup}[1]{% 
\vskip\skip\@nameuse{mp#1footins}\ifl@dpairing\ifparledgroup% 
\leavevmode\marks\parledgroup@{begin}\marks\parledgroup@series{#1}\marks\parledgroup@type{Xfootnote}%% 
\fi\fi\normalcolor\ifparledgroup% 
\ifl@dpairing% 
\else% 
\setXnoteswidthliketwocolumns@{#1}%% 
\setXnotespositionliketwocolumns@{#1}%% 
\print@Xfootnoterule{#1}%% 
\fi% 
\else% 
\setXnoteswidthliketwocolumns@{#1}%% 
\setXnotespositionliketwocolumns@{#1}%% 
\print@Xfootnoterule{#1}%% 
\fi%% 
\setlength{\parindent}{\opt}% 
\csuse{Xnotefontsize@#1}\csuse{txtbeforeXnotes@#1}}% 
\unvbox\csname mp#1footins\endcsname}

\footnormal We can now define all the parameters for the six series of footnotes; initially they use the ‘normal’ footnote formatting, which is set up by calling \footnormal. You can switch to another type of formatting by using \footparagraph, \foottwocol, or \footthreecol.

Switching to a variation of ‘normal’ formatting requires changing the quantities defined in \footnormal. The best way to proceed would be to make a copy of this macro, with a different name, make your desired changes in that copy, and then invoke it, giving it the letter of the footnote series you wish to control.
(We have not defined baseline skip values like \baselineskip, since this is one of the quantities set in \notefontsetup.) What we want to do here is to say something like the following for each footnote series. (This is an example, not part of the actual \eledmac code.)

\skip\Afootins=12pt plus5pt minus5pt
\count\Afootins=1000
\dimen\Afootins=0.8\vsize
\let\vAfootnote=\normalvfootnote \let\Afootfmt=\normalfootfmt
\let\Afootstart=\normalfootstart \let\Afootgroup=\normalfootgroup
\let\Afootnoterule=\normalfootnoterule

Instead of repeating ourselves, we define a \footnormal macro that makes all these assignments for us, for any given series letter. This also makes it easy to change from any different system of formatting back to the normal setting.

\ledfootinsdim Have a constant value for the \dimen\footins

\preXnotes@ If user redefines \preXnotes@, via \preXnotes to a value greater than 0 pt, this skip will be added before first series notes instead of the notes skip.

\preXnotes The same, but for familiar footnotes.

\preXnotes@ Now we set up the \footnormal macro itself. It takes one argument: the footnote series letter.

\newcommand*{\footnormal}[1]{%
Now do the setup for minipage footnotes. We use as much as possible of the normal setup as we can (so the notes will have a similar layout).

Some of these values deserve comment: the \texttt{\texttt{\texttt{dimen}}} setting allows 80\% of the page to be occupied by notes; the \texttt{\texttt{\texttt{skip}}} setting is deliberately flexible, since pages with lots of notes attached to many of the lines can be a bit hard for \TeX{} to make.

\subsection{Paragraphed footnotes}

The paragraphed-footnote option reformats all the footnotes of one series for a page into a single paragraph; this is especially appropriate when the notes are numerous and brief. The code is based on \textit{The TeXbook}, pp.398–400, with alterations for our environment. This algorithm uses a considerable amount of save-stack space: a \TeX{} of ordinary size may not be able to handle more than about 100 notes of this kind on a page.

\begin{command}{footparagraph}

The \texttt{\texttt{\texttt{\texttt{footparagraph}}}} macro sets up everything for one series of the footnotes so that they’ll be paragraphed; it takes the series letter as argument. We include the setting of \texttt{\texttt{\texttt{\texttt{count}}}\texttt{\texttt{\texttt{footins}}} to 1000 for the footnote series just in case you are switching to paragraphed footnotes after having columnar ones, since they change this value (see below).

It is important to call \texttt{\texttt{\texttt{\texttt{footparagraph}}}} only after \texttt{\texttt{\texttt{\texttt{\texttt{\texttt{hsize}}}}}} has been set for the pages that use this series of notes; otherwise \TeX{} will try to put too many or too few of these notes on each page. If you need to change the \texttt{\texttt{\texttt{\texttt{hsize}}}} within the document, call \texttt{\texttt{\texttt{\texttt{footparagraph}}}} again afterwards to take account of the new value. The argument of \texttt{\texttt{\texttt{\texttt{footparagraph}}}} is the letter (A–E) denoting the series of notes to be paragraphed.
And the extra setup for minipages.

\footfudgefiddle For paragraphed footnotes TEX has to estimate the amount of space required. If it underestimates this then the notes may get too long and run off the bottom of the text block. \footfudgefiddle can be increased from its default 64 (say to 70) to increase the estimate.

\para@footsetup \footparagraph calls the \para@footsetup macro to calculate a special fudge factor, which is the ratio of the \baselineskip to the \hsize. We assume that the proper value of \baselineskip for the footnotes (normally 9pt) has been set already, in \notefontsetup. The argument of the macro is again the note series letter.

Peter Wilson thinks that \columnwidth should be used here for L\TeX not \hsize. I've also included \footfudgefiddle.

EDMAC defines \en@number which does the same as the L\TeX \strip@pt, namely strip the characters pt from a dimen value. Eledmac use \strip@pt.

\para@footstart \para@footstart is the same as \normal@footstart, but we give it again to ensure that \rightskip and \leftskip are zeroed (this needs to be done before \para@footgroup in the output routine). You might have decided to change this for other kinds of note, but here it should stay as it is. The size of paragraphed notes is calculated using a fudge factor which in turn is based on \hsize. So the paragraph of notes needs to be that wide.

The argument of the macro is again the note series letter.

\newcommand*{\para@footstart}[1]{% 
\rightskip=0pt \leftskip=0pt \parindent=0pt}
### 25.5 Paragraphed footnotes

\paravariables{vfootnote} is a version of the \texttt{\vfootnote} command that’s used for paragraphed notes. It gets appended to the \texttt{\inserts@list} list by an outer-level footnote command like \texttt{\Afootnote}. The first argument is the note series letter; the second is the full text of the printed note itself, including line numbers, lemmata, and footnote text.

The initial model for this insertion is, of course, the \texttt{\insert\footins} definition in *The \TeX\ Book*, p. 398. There, the footnotes are first collected up in hboxes, and these hboxes are later unpacked and stuck together into a paragraph.

However, Michael Downes has pointed out that because text in hboxes gets typeset in restricted horizontal mode, there are some undesirable side-effects if you later want to break such text across lines. In restricted horizontal mode, where \TeX\ does not expect to have to break lines, it does not insert certain items like \texttt{\discretionary}. If you later unbox these hboxes and stick them together, as the \texttt{\TeX\ Book} macros do to make these footnotes, you lose the ability to hyphenate after an explicit hyphen. This can lead to overfull \texttt{\hbox}s when you would not expect to find them, and to the uninitiated it might be very hard to see why the problem had arisen.\footnote{Wayne Sullivan pointed out to us another subtle problem that arises from the same cause: \TeX\ also leaves the \texttt{\language} whatsit nodes out of the horizontal list. So changes from one language to another will not invoke the proper hyphenation rules in such footnotes. Since critical editions often do deal with several languages, especially in a footnotes, we really ought to get this bit of code right.}

Wayne Sullivan pointed out to us another subtle problem that arises from the same cause: \TeX\ also leaves the \texttt{\language} whatsit nodes out of the horizontal list.\footnote{See *The \TeX\ Book*, p. 455 (editions after January 1990).} To get around these problems, Wayne suggested emendations to the \texttt{\TeX\book} versions of these macros which are broadly the same as those described by Michael: the central idea (also suggested by Donald Knuth in a letter to Michael) is to avoid collecting the text in an \texttt{\hbox} in the first place, but instead to collect it in a \texttt{\vbox} whose width is (virtually) infinite. The text is therefore typeset in unrestricted horizontal mode, as a paragraph consisting of a single long line. Later, there is an extra level of unboxing to be done: we have to unpack the \texttt{\vbox}, as well as the hboxes inside it, but that’s not too hard. For details, we refer you to Michael’s
article, where the issues are clearly explained.\footnote{Wayne supplied his own macros to do this, but since they were almost identical to Michael’s, we have used the latter’s \texttt{unvxh} macro since it is publicly documented.} Michael’s unboxing macro is called \texttt{unvxh}: \texttt{un vbox}, extract the last line, and unhbox it.

Doing things this way has an important consequence: as Michael pointed out, you really can’t put an explicit line-break into a note built in a \texttt{vbox} the way we are doing.\footnote{Line Breaking’, p. 610.} In other words, be very careful not to say \texttt{break}, or \texttt{penalty-10000}, or any equivalent inside your para-footnote. If you do, most of the note will probably disappear. You are allowed to make strong suggestions; in fact \texttt{penalty-9999} will be quite okay. Just don’t make the break mandatory. We haven’t applied any of Michael’s solutions here, since we feel that the problem is exiguous, and \texttt{eledmac} is quite baroque enough already. If you think you are having this problem, look up Michael’s solutions.

One more thing: we set \texttt{leftskip} and \texttt{rightskip} to zero. This has the effect of neutralizing any such skips which may apply to the main text (cf.\textsuperscript{25.3} \pageref{p.130} above). We need to do this, since \texttt{footfudgefactor} is calculated on the assumption that the notes are \texttt{hsize} wide.

So, finally, here is the modified foot-paragraph code, which sets the footnote in vertical mode so that language and discretionary nodes are included.

\begin{verbatim}
\newcommand*{\para@vfootnote}[2]{
  \insert\csname #1footins\endcsname
  \bgroup
  \csuse{bhookXnote@#1}
  \csuse{Xnotefontsize@#1}
  \footsplitskips
  \setbox0=\vbox{\hsize=\maxdimen
    \noindent\csname #1footfmt\endcsname#2[#1]}
  \setbox0=\hbox{\unvxh0[#1]}
  \dp0=0pt
  \ht0=\csname #1footfudgefactor\endcsname\wd0

  Here we produce the contents of the footnote from box 0, and add a penalty of 0 between boxes in this insert.

  \if@RTL
    \noindent \leavevmode\fi
  \penalty0
  \egroup}
\end{verbatim}

The final penalty of 0 was added here at Wayne’s suggestion to avoid a weird page-breaking problem, which occurs on those occasions when \TeX\ attempts to split foot paragraphs. After trying out such a split (see \textit{The \TeX\book}, p. 124), \TeX\ inserts a penalty of \texttt{-10000} here, which nearly always forces the break at the end of the whole footnote paragraph (since individual notes can’t be split) even when this leads to an overfull vbox. The change above results in a penalty of 0 instead which allows, but doesn’t force, such breaks. This penalty of 0 is later removed, after page breaks have been decided, by the \texttt{\unpenalty} macro in \texttt{\makehboxofhboxes}. So it does not affect how the footnote paragraphs are
typeset (the notes still have a penalty of −10 between them, which is added by \parafootfmt).

\texttt{\mppara@vfootnote This version is for minipages.}

\begin{verbatim}
2130 \newcommand*{\mppara@vfootnote}[2]{% 2131  \global\setbox\nameuse{mp#1footins}\vbox{% 2132  \unvbox\nameuse{mp#1footins}% 2133  \csuse{bhookXnote@#1}% 2134  \csuse{Xnotefontsize@#1}% 2135  \footsplitskips% 2136  \setbox0=vbox\hspace=\maxdimen% 2137  \noindent\color@begingroup\csname #1footfmt\endcsname #2[#1]\color@endgroup} 2138  \setbox0=\hbox{\unvxh0[#1]}% 2139  \dp0=\z@% 2140  \ht0=\csname #1footfudgefactor\endcsname\wd0% 2141  \box0% 2142  \penalty0% 2143 }% 2144
\end{verbatim}

\texttt{\unvxh Here is (modified) Michael’s definition of \unvxh, used above. Michael’s macro also}

takes care to remove some unwanted penalties and glue that \TeX{} automatically
attaches to the end of paragraphs. When \TeX{} finishes a paragraph, it throws away
any remaining glue, and then tacks on the following items: a \penalty of 10000,
a \parfillskip and a \rightskip (The \TeX{}book, pp. 99–100). \unvxh cancels
these unwanted paragraph-final items using \unskip and \unpenalty.

\begin{verbatim}
2145 \newcommandx*{\unvxh}[2=Z]{% 2146  \setbox0=\vbox{\unvbox#1% 2147  \global\setbox1=\lastbox}% 2148  \unhbox1 2149  \unskip% 2150  \unskip% 2151  \unpenalty% 2152  \hskip\csuse{afternote@#2}}% 2153 \end{verbatim}

\texttt{\parafootfmt \parafootfmt is \normalfootfmt adapted to do the special stuff needed for para-
graphed notes—leaving out the \endgraf at the end, sticking in special penalties
and kern, and leaving out the \footstrut. The first argument is the line and
page number information, the second is the lemma, the third is the text of the
footnote, and the fourth is the series (optional, for backward compatibility).

\begin{verbatim}
2154 \newcommandx*{\parafootfmt}[4=Z]{% 2155  \insertparafootsep{#4}% 2156  \Xledsetnormalparstuff{#4}% 2157  \printlinefootnote{#1}{#4}% 2158  {\nottoggle{Xlemmadisablefontselection@#4}\{\select@lemmafont#1|#2\}}% 2159  {\iftoggle{nosep@}{\hskip\csuse{inplaceoflemmaseparator@#4}}% 2160  {\nobreak\hskip\csuse{beforelemmaseparator@#4}\csuse{lemmaseparator@#4}\hskip\csuse{afterlemmaseparator@#4}%} 2161}
\end{verbatim}
Note that in the above definition, the penalty of $-10$ encourages a line break between notes, so that notes have a slight tendency to begin on new lines. The \texttt{\insertparafootsep} command is used to insert the \texttt{\parafootsep@series} between each note in the same page.

This \texttt{footgroup} code is modelled on the macros in \textit{The \TeX \book}, p. 399. The only difference is the \texttt{\unpenalty} in \texttt{\makeboxofhboxes}, which is there to remove the penalty of 0 which was added to the end of each footnote by \texttt{\para@vfootnote}.

The call to \texttt{\notefontsetup} is to ensure that the correct \texttt{\baselineskip} for the footnotes is used. The argument is the note series letter.

The minipage version.

\begin{verbatim}
\newcommand*{\mppara@footgroup}[1]{% 
  \setXnoteswidthliketwocolumns@{#1}%
  \vskip\skip\@nameuse{mp#1footins}%
  \ifl@dpairing\ifparledgroup% 
    \leavevmode\marks\parledgroup@{begin}%
    \marks\parledgroup@series{#1}%
    \marks\parledgroup@type{Xfootnote}%
  \fi\fi
  \normalcolor
  \ifparledgroup%
    \ifl@dpairing% 
      \else%
        \setXnoteswidthliketwocolumns@{#1}%
        \setXnotespositionliketwocolumns@{#1}%
        \print@Xfootnoterule{#1}%%
      \fi%
    \else%
      \setXnoteswidthliketwocolumns@{#1}%
      \setXnotespositionliketwocolumns@{#1}%
      \print@Xfootnoterule{#1}%
    \fi%
  \else% 
    \setXnoteswidthliketwocolumns@{#1}%
    \setXnotespositionliketwocolumns@{#1}%
    \print@Xfootnoterule{#1}%
  \fi% 
  \unvbox\csname mp#1footins\endcsname
  \ifcsstring{Xragged@#1}{L}{\RaggedLeft}{}%
  \ifcsstring{Xragged@#1}{R}{\RaggedRight}{}%
  \makeboxofhboxes
\end{verbatim}
25.6 Columnar footnotes

The command `\insertparafootsep{⟨series⟩}` must be called at the beginning of `\parafootftm` (and like commands).

We will now define macros for three-column notes and two-column notes. Both sets of macros will use `\rigidbalance`, which splits a box (#1) into a number (#2) of columns, each with a space (#3) between the top baseline and the top of the `\vbox`. The `\rigidbalance` macro is taken from *The TeXbook*, p. 397, with a slight change to the syntax of the arguments so that they don't depend on white space. Note also the extra unboxing in `\splitoff`, which allows the new `\vbox` to have its natural height as it goes into the alignment.

The LATEX `\line` macro has no relationship to the TeX `\line`. The LATEX equivalent is `\@@line`.
25.6.1 Three columns

\footthreecol You say $\texttt{\footthreecol A}$ to have the A series of the footnotes typeset in three columns. It is important to call this only after \hsize has been set for the document.

\threecolfootsetup The \threecolfootsetup macro calculates and sets some numbers for three-column footnotes.

We set the \count of the foot insert to 333. Each footnote can be thought of as contributing only one third of its height to the page, since the footnote insertion has been made as a long narrow column, which then gets trisected by the \rigidbalance routine (inside \threecolfootgroup). These new, shorter columns are saved in a box, and then that box is put back into the footnote insert,
replacing the original collection of the footnotes. This new box is, therefore, only about a third of the height of the original one.

The \texttt{dimen} value for this note series has to change in the inverse way: it needs to be three times the actual limit on the amount of space these notes are allowed to fill on the page, because when \TeX is accumulating material for the page and checking that limit, it doesn’t apply the \texttt{count} scaling.

\begin{verbatim}
\newcommand*{\threecolfootsetup}[1]{% 
  \count\texttt{csname #1footins}333 
  \csxdef{default@#1footins}{333}%Use this to confine the notes to one side only 
  \multiply\texttt{dimen}\texttt{csname #1footins}\texttt{endsname} \thr@@}
\end{verbatim}

\texttt{\threecolvfootnote} \texttt{\threecolvfootnote} is the \texttt{\footnote} command for three-column notes. The call to \texttt{\notefontsetup} ensures that the \texttt{\splittopskip} and \texttt{\splitmaxdepth} take their values from the right \texttt{\strutbox}: the one used in a footnotes. Note especially the importance of temporarily reducing the \texttt{\hsize} to 0.3 of its normal value. This determines the widths of the individual columns. So if the normal \texttt{\hsize} is, say, 10 cm, then each column will be $0.3 \times 10 = 3$ cm wide, leaving a gap of 1 cm spread equally between columns (i.e., .5 cm between each).

The arguments are 1) the note series letter and 2) the full text of the note (including numbers, lemma and text).

\begin{verbatim}
\notbool{parapparatus@}{\newcommand*}{\newcommand}{\threecolfootnote}[2]{% 
  \insert\texttt{csname #1footins}\texttt{endsname}\bgroup 
  \csuse{Xnotefontsize@#1} 
  \footsplitskips 
  \texttt{csname #1footfmt}\texttt{#2}[#1]\egroup}
\end{verbatim}

\texttt{\threecolfootfmt} \texttt{\threecolfootfmt} is the command that formats one note. It uses \texttt{\raggedright}, which will usually be preferable with such short lines. Setting the \texttt{\parindent} to zero means that, within each individual note, the lines begin flush left.

The arguments are 1) the line numbers, 2) the lemma and 3) the text of the \texttt{-footnote} command 4) optional (for backward compatibility): the series.

\begin{verbatim}
\notbool{parapparatus@}{\newcommand*}{\newcommand}{\threecolfootfmt}[4][4=Z]{% 
  \normal@pars 
  \hsize \csuse{hsizethreecol@#4} 
  \nottoggle{Xparindent@#4}{\parindent=\z@}{\parindent=} 
  \tolerance=5000 
  \hangindent=\csuse{Xhangindent@#4} 
  \leavevmode 
  \csuse{Xcolalign@#4} 
  \strut{\printlinefootnote{#1}{#4}} 
  \{\nottoggle{Xlemmadisablefontselection@#4}{\select@lemmafont#1|#2}{#2}}%}
\end{verbatim}
And here is the \footgroup macro that’s called within the output routine to re-group the notes into three columns. Once again, the call to \notefontsetup is there to ensure that it is the right \splittopskip—the one used in footnotes—which is used to provide the third argument for \rigidbalance. This third argument (\@h) is the \topskip for the box containing the text of the footnotes, and does the job of making sure the top lines of the columns line up horizontally. In The \TeXbook, p. 398, Donald Knuth suggests retrieving the output of \rigidbalance, putting it back into the insertion box, and then printing the box. Here, we just print the \line which comes out of \rigidbalance directly, without any re-boxing.

The setup for minipages.

25.6 Columnar footnotes

25.6.2 Two columns

```
\footnotewcol \textbf{You say} \footnotewcol\{A\} \textbf{to have the A series of the footnotes typeset in two columns. It is important to call this only after} \hsize \textbf{has been set for the document.}
```

```
\newcommand*{\footnotewcol}[1]{%
  \csgdef{series@display#1}{twocol}
  \expandafter\let\csname v#1footnote\endcsname=\twocolvfootnote
  \expandafter\let\csname #1footfmt\endcsname=\twocolfootfmt
  \expandafter\let\csname #1footgroup\endcsname=\twocolfootgroup
  \dimen\csname #1footins\endcsname=\csuse{maxhXnotes@#1}\
  \skip\csname #1footins\endcsname=\csuse{beforeXnotes@#1}\
  \advance\skip\csname #1footins\endcsname by\csuse{afterXrule@#1}\
  \twocolfootsetup{#1}
}
```

```
\twocolfootsetup \twocolvfootnote \twocolfootfmt \twocolfootgroup \textbf{Here is a series of macros which are very similar to their three-column counterparts.}
```

```
\newcommand*{\twocolfootsetup}[1]{%
  \count\csname #1footins\endcsname 500
  \csxdef{default@#1footins}{500}%Use this to confine the notes to one side only
  \multiply\dimen\csname #1footins\endcsname \tw@
  \notbool{parapparatus@}{\newcommand}{\newcommand}{\twocolvfootnote}[2]{\insert\csname #1footins\endcsname\bgroup
  \csuse{Xnotefontsize@#1}
  \footsplitskips
  \csname #1footfmt\endcsname #2[#1]\egroup}
  \notbool{parapparatus@}{\newcommandx*}{\newcommandx}{\twocolfootfmt}[4][4=Z]{{% 4th arg is optional, for backward compatibility
    \normal@pars
    \hsize \csuse{hsizetwocol@#4}
    \nottoggle{Xparindent@#4}{\parindent=z0}{}
    \tolerance=5000
    \hangindent=\csuse{Xhangindent@#4}
    \leavevmode
    \csuse{Xcolalign@#4}%
    \strut\printlinefootnote{#1}{#4}%%%%%%%%%%%%%%%%%%%%%%%
    \nottoggle{Xllemmaseparator@#4}{\select@lemmafont#1|#2}{#2}%%%%%%%%%%%%%%%%%%%%%%%
    \iftoggle{nosep@}{\hskip\csuse{inplaceoflemmaseparator@#4}}{% %
      \ifcsempty{lemmaseparator@#4}{\hskip\csuse{inplaceoflemmaseparator@#4}}%
      \hskip\csuse{inplaceoflemmaseparator@#4}}%
}}
```

```
\twocolfootsetup \twocolvfootnote \twocolfootfmt \twocolfootgroup \textbf{In this case, each note is assumed to contribute only a half a line of text. And the notes are set in columns giving a gap between them of one tenth of the} \hsize.\textbf{\}
```

```
```

26 Familiar footnotes

26.1 Generality

The original EDMAC provided users with five series of critical footnotes (\Afootnote \Bfootnote \Cfootnote \Dfootnote \Efootnote), and \LaTeX provides a single numbered footnote. The \eledmac package uses the EDMAC mechanism to provide six series of numbered footnotes.
First, though, the `footmisc` package has an option whereby two or more consecutive \footnotes have their marks separated by commas. This seems such a useful ability that it is provided automatically by \texttt{eledmac}.

These macros may have been defined by the \texttt{memoir} class, are provided by the \texttt{footmisc} package and perhaps by other footnote packages.

A pair of self-cancelling kerns. This may have been defined in the \texttt{memoir} class.

We have to modify \texttt{@footnotetext} and \texttt{@footnotemark}. However, if \texttt{memoir} is used the modifications have already been made.

Add \texttt{@m@mmf@prepare} at the end of \texttt{@footnotetext}. Modify \texttt{@footnotemark} to cater for adjacent \footnotes.

Finished the modifications for the non-memoir case.
In order to enable the regular \footnotes in numbered text we have to play around with its \@footnotetext, using different forms for when in numbered or regular text.

\l@dbfnote \l@dbfnote adds the footnote to the insert list, and \vl@dbfnote calls the original \@footnotetext.

26.2 Footnote formats

Some of the code for the various formats is remarkably similar to that in section 25.3.

The following macros generally set things up for the ‘standard’ footnote format.
The `\normalbodyfootmarkX{\langle series \rangle}` really typesets the in-text marker. The style is the normal superscript.

`\newcommand*{\normalbodyfootmarkX}[1]{%  \hbox{\textsuperscript{\normalfont@nameuse{@thefnmark#1}}}}`  

`\normalvfootnoteX{\langle series \rangle}{\langle text \rangle}` does the `\insert` for the `\langle series \rangle` and calls the series' `\footfmt...` to format the `\langle text \rangle`.

`\notbool{parapparatus@}{\newcommand*}{\newcommand}{\normalvfootnoteX}[2]{%  \insert@nameuse{footins#1}\bgroup  \csuse{bhooknoteX@#1}  \csuse{notefontsizeX@#1}  \ifl@dpairing\ifl@dpaging\else%  \setnotesXwidthliketwocolumns@{#1}%  \fi\fi%  \setnotesXpositionliketwocolumns@{#1}%  \spaceskip=z@skip \xspaceskip=z@skip  \csuse{\csuse{footnote@dir}}@nameuse{footfmt#1}{#1}{#2}\egroup}`

The minipage version.

`\mpnormalvfootnoteX{\langle series \rangle}{\langle text \rangle}`

The `\normalfootfmtX{\langle series \rangle}{\langle text \rangle}` typesets the footnote text, prepended by the marker.

`\notbool{parapparatus@}{\newcommand*}{\newcommand}{\normalfootfmtX}[2]{%  \ifluatex%  \luatextextdir\footnote@luatextextdir%  \luatexpardir\footnote@luatexpardir%  \par%  \fi%  \protected@edef\@currentlabel{%  \@nameuse{@thefnmark#1}%  }%  }%  }%  \ledsetnormalparstuffX{#1}%  \hangindent=\csuse{hangindentX@#1}%  {{\csuse{notenumfontX@#1}@nameuse{footfootmark#1}}\strut\par}`
\normalfootfootmarkX \normalfootfootmarkX{⟨series⟩} is called by \normalfootfmtX to typeset the footnote marker in the footer before the footnote text.

\newcommand*{\normalfootfootmarkX}[1]{% #2\strut\par}}
\textsuperscript{\@nameuse{@thefnmark#1}}}

\normalfootstartX \normalfootstartX{⟨series⟩} is the ⟨series⟩ footnote starting macro used in the output routine.
\newcommand*{\normalfootstartX}[1]{% #2\strut\par}}
\ifdimequal{0pt}{\prenotesX@}{}%
\iftoggle{prenotesX@}{% #2\strut\par}}
\togglefalse{prenotesX@}%
\skip\csname footins#1\endcsname=%
\dimexpr\csuse{prenotesX@}+\csuse{afterruleX@#1}\relax%
\}%
\}%
\leftskip=\z@%
\rightskip=\z@%
\ifl@dpairing\else%
\hsize=\old@hsize%
\fi%
\setnotesXwidthliketwocolumns@{#1}%
\setnotesXpositionliketwocolumns@{#1}%
\print@footnoteXrule{#1}%
\}%

\normalfootnoteruleX The rule drawn before the footnote series group.
\let\normalfootnoteruleX=\footnoterule

\normalfootgroupX \normalfootgroupX{⟨series⟩} sends the contents of the ⟨series⟩ insert box to the output page without alteration.
\newcommand*{\normalfootgroupX}[1]{% #2\strut\par}}
\unvbox\@nameuse{footins#1}%
\hsize=\old@hsize%
}\}%

\mpnormalfootgroupX The minipage version.
\newcommand*{\mpnormalfootgroupX}[1]{% #2\strut\par}}
\vskip\skip\@nameuse{mpfootins#1}
\ifl@dpairing\ifparledgroup%
Footnote formats

\footnormalX \footnormalX{(series)} initialises the settings for the \texttt{(series)} footnotes. This should always be called for each series.
The following macros set footnotes in two columns. It is assumed that the length of each footnote is less than the column width.

\foottwocolX \foottwocolX{(series)}

The following macros set footnotes in two columns. It is assumed that the length of each footnote is less than the column width.

\foottwocolX \foottwocolX{(series)}

The following macros set footnotes in two columns. It is assumed that the length of each footnote is less than the column width.

\foottwocolX \foottwocolX{(series)}

The following macros set footnotes in two columns. It is assumed that the length of each footnote is less than the column width.

\foottwocolX \foottwocolX{(series)}

The following macros set footnotes in two columns. It is assumed that the length of each footnote is less than the column width.

\foottwocolX \foottwocolX{(series)}

The following macros set footnotes in two columns. It is assumed that the length of each footnote is less than the column width.

\foottwocolX \foottwocolX{(series)}

The following macros set footnotes in two columns. It is assumed that the length of each footnote is less than the column width.

\foottwocolX \foottwocolX{(series)}
\twocolfootsetupX \twocolfootsetupX{(series)}
\mptwocolfootsetupX
\newcommand*{\twocolfootsetupX}{[1]{\count\csname footins#1\endcsname 500}}
\newcommand*{\mpttwocolfootsetupX}{[1]{\count\csname mpfootins#1\endcsname 500}}
\twocolvfootnoteX \twocolvfootnoteX{(series)}
\notbool{parapparatus@}{\newcommand*}{\newcommand}{\twocolvfootnoteX}{[2]{\insert\csname footins#1\endcsname\bgroup}}
\twocolfootfmtX \twocolfootfmtX{(series)}
\notbool{parapparatus@}{\newcommand*}{\newcommand}{\twocolfootfmtX}{[2]{\protected@edef\@currentlabel{\@nameuse{@thefnmark#1}}\normal@pars}}
\twocolfootgroupX \twocolfootgroupX{(series)}
\mptwocolfootgroupX\newcommand*{\twocolfootgroupX}{[1]{\{\csuse{notefontsizeX@#1}\splittopskip=\ht\strutbox}}
\newcommand*{\mpttwocolfootgroupX}{[1]{\vskip\skip\@nameuse{mpfootins#1}}}
26.4 Three columns footnotes

The following macros set footnotes in three columns. It is assumed that the length of each footnote is less than the column width.

\footthreecolX \footthreecolX{⟨series⟩}

\newcommand*{\footthreecolX}{⟨series⟩}{% 
\csdef{series@displayX}{threecolX} 
\expandafter\let\csname regvfootnote#1\endcsname=\threecolvfootnoteX 
\expandafter\let\csname footfmt#1\endcsname=\threecolfootfmtX 
\expandafter\let\csname footgroup#1\endcsname=\threecolfootgroupX 
\dimen\csname footins#1\endcsname=\csuse{maxhnotesX@#1} 
\skip\csname footins#1\endcsname=\csuse{beforenotesX@#1} 
\advance\skip\csname footins#1\endcsname by \csuse{afterruleX@#1}\relax 
\threecolfootsetupX{#1} 
\ifnoledgroup\else% 
\expandafter\let\csname mpvfootnote#1\endcsname=\mpnormalvfootnoteX 
\expandafter\let\csname mpfootgroup#1\endcsname=\mpthreecolfootgroupX 
\skip\csname mpfootins#1\endcsname=\csuse{beforenotesX@#1} 
\advance\skip\csname mpfootins#1\endcsname by \csuse{afterruleX@#1} 
\mpthreecolfootsetupX{#1} 
\fi% 
}

\threecolfootsetupX \threecolfootsetupX{⟨series⟩}
\mpthreecolfootsetupX

\newcommand*{\threecolfootsetupX}[1]{% 
\count\csname footins#1\endcsname=333 
\multiply\dimen\csname footins#1\endcsname by \thr@@ 
\newcommand*{\mpthreecolfootsetupX}[1]{% 
\count\csname mpfootins#1\endcsname=333 
\multiply\dimen\csname mpfootins#1\endcsname by \thr@@ 
}
26.4 Three columns footnotes

\threecolvfootnoteX \threecolvfootnoteX{⟨series⟩}{⟨text⟩}

\notbool{parapparatus@}{\newcommand*}{\newcommand}{\threecolvfootnoteX}[2]{% 
\insert\csname footins#1\endcsname\bgroup
\csuse{notefontsizeX@#1}
\footspitskips
\@nameuse{footfmt#1}{#1}{#2}\egroup
}

\threecolfootfmtX \threecolfootfmtX{⟨series⟩}
\notbool{parapparatus@}{\newcommand*}{\newcommand}{\threecolfootfmtX}[2]{%
\protected@edef\@currentlabel{\@nameuse{@thefnmark#1}}%
\hangindent=\csuse{hangindentX@#1}%
\normal@pars
\hsize \csuse{hsizethreecolX@#1}
\nottoggle{parindentX@#1}{\parindent=\z@}{}%
\tolerance=5000\relax
\leavevmode
\csuse{colalignX@#1}%
{\csuse{notenumfontX@#1}\@nameuse{footfootmark#1}\strut#2\strut\par}
}

\threecolfootgroupX \threecolfootgroupX{⟨series⟩}
\mpthreecolfootgroupX \mpthreecolfootgroupX{⟨series⟩}
\newcommand*{\threecolfootgroupX}[1]{{\csuse{notefontsizeX@#1}\
\splittopskip=\ht\strutbox\expandafter\rigidbalance\csname footins#1\endcsname \thr@@ \splittopskip}}
\newcommand*{\mpthreecolfootgroupX}[1]{{\vskip\skip\@nameuse{mpfootins#1}\ifl@dpairing\ifparledgroup
\leavevmode\marks\parledgroup@{begin}\marks\parledgroup@series{#1}\marks\parledgroup@type{footnoteX}\fi\else\setnotesXwidthliketwocolumns@{#1}\setnotesXpositionliketwocolumns@{#1}\print@footnoteXrule{#1}\fi\else\setnotesXwidthliketwocolumns@{#1}\setnotesXpositionliketwocolumns@{#1}\print@footnoteXrule{#1}\fi\splittopskip=\ht\strutbox}}
26.5 Paragraphed footnotes

The following macros set footnotes as one paragraph.

\footparagraphX \footparagraphX{(series)}
\newcommand*{\footparagraphX}{[1]{\csname noteftontsizeX@#1\endcsname %
\coskip\csname prevpage#1\endcsname %
\expandafter\newcount\csname prevpage#1\endcsname %
\expandafter\let\csname footstart#1\endcsname=\parafootstartX %
\expandafter\let\csname regvfootnote#1\endcsname=\para@vfootnoteX %
\expandafter\let\csname footfmt#1\endcsname=\parafootfmtX %
\expandafter\let\csname footgroup#1\endcsname=\para@footgroupX %
\expandafter\let\csname footnoterule#1\endcsname=\normalfootnoteruleX %
\count\csname footins#1\endcsname=1000 %Use this to confine the notes to one side only
\coskip\csname footins#1\endcsname=\csuse{maxhnotesX@#1} %
\advance\coskip\csname footins#1\endcsname by\csuse{afterruleX@#1} %
\para@footsetupX{#1} %
\ifnoledgroup@\else %
\expandafter\let\csname mpvfootnote#1\endcsname=mppara@vfootnoteX %
\expandafter\let\csname mpfootgroup#1\endcsname=mppara@footgroupX %
\count\csname mfootins#1\endcsname=1000 %
\coskip\csname mfootins#1\endcsname=\csuse{maxhnotesX@#1} %
\advance\coskip\csname mfootins#1\endcsname by\csuse{afterruleX@#1} %
\fi %
}

\para@footsetupX \para@footsetupX{(series)}
\newcommand*{\para@footsetupX}{[1]{\csuse{notefontsizeX@#1} %
\setnotesXwidthliketwocolumns@{#1} %
\dimen0=\baselineskip %
\multiply\dimen0 by 1024 %
\divide\dimen0 by \columnwidth %
\multiply\dimen0 by \footfudgefiddle\relax %
\expandafter %
\xdef\csname footfudgefactor#1\endcsname= %
\expandafter\strip@pt\dimen0 )}}

\parafootstartX \parafootstartX{(series)}
\newcommand*{\parafootstartX}{[1]{\csuse{notefontsizeX@#1} %
\ifdimequal{0pt}{\prenotesX@}{} %
\expandafter %
\xdef\csname footfudgefactor#1\endcsname= %
\expandafter\strip@pt\dimen0 )}}
26.5 Paragraphed footnotes

\iftoggle{prenotesX@}{
\skip\csname footins#1\endcsname=%
\dimexpr\csuse{prenotesX@}+\csuse{afterruleX@#1}\relax%
}
{}%
\vskip\skip\csname footins#1\endcsname%
\leftskip=\z@
\rightskip=\z@
\parindent=\z@
\vskip\skip\@nameuse{footins#1}%
\setnotesXwidthliketwocolumns@{#1}%
\setnotesXpositionliketwocolumns@{#1}%
\print@footnoteXrule{#1}%
}

\para@vfootnoteX \para@vfootnoteX{(series)}{(text)}
\mppara@vfootnoteX
\newcommand*{\para@vfootnoteX}[2]{%
\insert\csname footins#1\endcsname
\begin{group}
\csuse{bhooknoteX@#1}
\csuse{notefontsizeX@#1}
\footnoteskips
\setbox0=vbox{\hsize=\maxdimen
\noindent\@nameuse{footfmt#1}{#1}{#2}}%
\setbox0=hbox{\unvxh0[#1]}%
\dp0=\z@
\ht0=\csname footfudgefactor#1\endcsname\wd0
\box0
\penalty0}
\newcommand*{\mppara@vfootnoteX}[2]{%
\global\setbox\@nameuse{mpfootins#1}vbox{%
\unvbox\@nameuse{mpfootins#1}
\begin{group}
\csuse{bhooknoteX@#1}
\csuse{notefontsizeX@#1}
\footnoteskips
\setbox0=vbox{\hsize=\maxdimen
\noindent\@nameuse{footfmt#1}{#1}{#2}\color@endgroup}}%
\setbox0=hbox{\unvxh0[#1]}%
\dp0=\z@
\ht0=\csname footfudgefactor#1\endcsname\wd0
\box0
\penalty0}

\parafootfmtX \parafootfmtX{(series)}
\newcommand*{\parafootfmtX}[2]{%  
protected@edef\@currentlabel{%  
@nameuse{thefnmark#1}%  
}%  
\insertparafootsep{#1}%  
\ledsetnormalparstuffX{#1}{%  
\csuse{notenumfontX@#1}\csuse{notenumfontX@#1}@nameuse{footfootmark#1}\strut%  
#2\penalty-10}}%  
\para@footgroupX \para@footgroupX{(series)}%  
\mppara@footgroupX \newcommand*{\para@footgroupX}[1]{%  
\unbox\csname footins#1\endcsname%  
\ifcsstring{raggedX@#1}{L}{\RaggedLeft}{}%  
\ifcsstring{raggedX@#1}{R}{\RaggedRight}{}%  
\makehboxofhboxes%  
\setbox0=\hbox{\unhbox0 \removehboxes}%  
\csuse{notefontsizeX@#1}%  
\noindent\unhbox0\par}  
\newcommand*{\mppara@footgroupX}[1]{{%  
\setnotesXwidthliketwocolumns@{#1}%  
\vskip\skip\@nameuse{mpfootins#1}%  
\ifl@dpairing\ifparledgroup  
\leavevmode%  
\marks\parledgroup@{begin}%  
\marks\parledgroup@series{#1}%  
\marks\parledgroup@type{footnoteX}%  
\fi\fi\normalcolor%  
\ifparledgroup%  
\ifl@dpairing%  
\else%  
\setnotesXwidthliketwocolumns@{#1}%  
\setnotesXpositionliketwocolumns@{#1}%  
\print@footnoteXrule{#1}%  
\fi%  
\else%  
\setnotesXwidthliketwocolumns@{#1}%  
\setnotesXpositionliketwocolumns@{#1}%  
\print@footnoteXrule{#1}%  
\fi%  
\fi%  
\unbox\csname mfootins#1\endcsname%  
\ifcsstring{raggedX@#1}{L}{\RaggedLeft}{}%  
\ifcsstring{raggedX@#1}{R}{\RaggedRight}{}%  
\makehboxofhboxes%  
\setbox0=\hbox{\unhbox0 \removehboxes}%  
\csuse{notefontsizeX@#1}%  
\noindent\unhbox0\par}
27 Footnotes’ width for two columns

We define here some commands which make sense only with \texttt{eledpar}, but must be called when defining notes parameters. These commands change the width of block notes to allow them to have the same size than two parallel columns.

These two commands are called at the beginning of critical or familiar notes groups. They set, if the option is enabled, the \texttt{\hsize}. They are also called at the on the setup for paragraphed notes.

\begin{verbatim}
\old@hsize
\setXnoteswidthliketwocolumns@
\setnotesXwidthliketwocolumns@
\end{verbatim}

These two commands set the position of the critical / familiar footnotes, depending on the hooks \texttt{Xnoteswidthliketwocolumns} and \texttt{notesXwidthliketwocolumns}. They call commands which are defined only in \texttt{eledpar}, because this feature has no sens without \texttt{eledpar}.

\begin{verbatim}
\setXnotespositionliketwocolumns@
\setnotesXpositionliketwocolumns@
\end{verbatim}
28 Footnotes’ order

\fnpos \footnotes The \fnpos and \mpfnpos simply place their arguments in \@fnpos and \@mpfnpos, which will be used later in the output routine.
\@fnpos \def\@fnpos{familiar-critical}
\@mpfnpos \def\@mpfnpos{critical-familiar}
\newcommand{\fnpos}[1]{\xdef\@fnpos{#1}}
\newcommand{\mpfnpos}[1]{\xdef\@mpfnpos{#1}}

29 Footnotes’ rule

Because the footnotes’ rules can be shifted to the right when footnotes are set like two columns, we don’t print them directly, but we put them in a \vbox.
\print@Xfootnoterule \print@footnoteXrule \newcommand{\print@Xfootnoterule}[1]{% \vskip-\csuse{afterXrule#1}\nointerlineskip\moveleft-\leftskip\vbox{\csuse{#1footnoterule}}\nointerlineskip\vskip\csuse{afterXrule#1}}%
\newcommand{\print@footnoteXrule}[1]{% \vskip-\csuse{afterruleX#1}\nointerlineskip\moveleft-\leftskip\vbox{\csuse{footnoterule#1}}\nointerlineskip\vskip\csuse{afterruleX#1}}%

30 Specific skip for first series of footnotes
\beforeXnotes insert a specific skip for the first series of notes in a page. As we can know in advance which series will be the first, we call \prepare@preXnotes before inserting any critical notes, in order to prevent page number overlapping.
1. If it is the first note of the current page, it changes the footnote skip for the
series to the value specified to `\beforeXnotes`. Keeps the series of the note
as the first one of the current page.

2. If it is not the first note of the current page:
   - If the current series is printed after the series kept as the first of the
current page, then nothing happens.
   - If the current series is printed before the series kept as the first of the
current page, then it changes the footnote of the current series to the
value normally used by the series which was marked as the first of the
page. Keeps the current series as the new first one of the current page.

For example, suppose the series order is A,B. We call first a `\Bfootnote`
and a `\Afootnote`. The only skips used are, finally, the skip specific to the first series
of the page, and the skip for the B series. If we have not called `\Afootnote`, the
only skip used is the skip specific to the first series of the page.

That is perfect.

The series skip and the first series of the current page are reset before the
footnotes are printed. Then, the footstart macros manage the problem of the first
series of the page.

After the rule, the space which is defined by `\afterXrule` does not depend on
whether the series is the first one of the page or not. So we use its normal value
for each series.

```latex
\firstXseries@ \prepare@preXnotes
```

The same thing is required for familiar notes and `\prenotesX`. 
31 Footnotes’ output

We have to add all the new kinds of familiar footnotes to the output routine. These are the class 1 feet. The normal way to add one series. \print@notes is replaced by \eledpar when using \Pages.

\newcommand{\print@notes}[1]{%}
\csuse{footstart#1}{#1}%
\csuse{footgroup#1}{#1}%
\print@notesX{#1}%
}

We print all the series of notes by looping on them. We check before printing them that they are not voided.

\newcommand*{\doxtrafeeti}{%}
\unless\ifnofamiliar%
\gdef\firstseriesX@{}%
\setbox\@outputbox \vbox{%}
\unvbox\@outputbox%
\def\do###1{%
\ifvoid\csuse{footins###1}\else%
\global\skip\csuse{footins###1}=%\csuse{beforenotesX###1}%
\global\advance\skip\csuse{footins###1} by\csuse{afterruleX###1}%
\print@notesX{###1}%
\fi%
\def\do#1{}
\do##1{%
\ifseriesbefore#1{\firstseriesX@}%
{%}
\global\skip\csuse{footins##1}=%\csuse{beforenotesX###1}-%\firstseriesX@%
\global\advance\skip\csuse{footins##1} by\csuse{afterruleX###1}%
\gdef\firstXseries@{###1}%
{%}
\do##1%
\do##1%
\do##1%
}
\dolistloop{\@series}}%
32 Endnotes

First, check the noend option.

\ifbool{noend@}{%\%Used instead of \ifnoend@ to prevent expansion problem

Endnotes of all varieties are saved up in a file, typically named ⟨jobname⟩.end.  
\ifl@dend@ is the output stream number for this file, and \ifl@dend@ is a flag that’s  
true when the file is open.

\l@dend@false 3014 \newwrite\l@dend@  
\newif\ifl@dend@  
\l@dend@true 3015 \newif\ifl@dend@

\l@dend@open  \l@dend@open and \l@dend@close are the macros that are used to open and close  
\l@dend@close the endnote file. Note that all our writing to this file is \immediate: all page and
line numbers for the endnotes are generated by the same mechanism we use for the footnotes, so that there’s no need to defer any writing to catch information from the output routine.

\begin{verbatim}
\newcommand{\l@dend@open}[1]{\global\l@dend@true\immediate\openout\l@d@end=#1\relax}
\newcommand{\l@dend@close}{\global\l@dend@false\immediate\closeout\l@d@end}
\l@dend@stuff
\l@dend@stuff is used by \begin{numbering} to do everything that’s necessary for the endnotes at the start of each section: it opens the \l@d@end file, if necessary, and writes the section number to the endnote file.
\newcommand{\l@dend@stuff}{\if\l@dend@\relax\else\l@dend@open{\jobname.end}\fi\immediate\write\l@d@end{\string\l@d@section{\the\section@num}}}\end{verbatim}

The \endprint here is nearly identical in its functioning to \texttt{\normalfootfmt}.

The endnote file also contains \l@d@section commands, which supply the section numbers from the main text; standard \texttt{eledmac} does nothing with this information, but it’s there if you want to write custom macros to do something with it. Arguments are:

- #1 Line numbers and font selection.
- #2 Lemma.
- #3 Note content.
- #4 Series.
- #5 Optional argument of \texttt{\Xendnote}.

\begin{verbatim}
\global\notbool{parapparatus@}{\long}\def\endprint#1#2#3#4#5{{\if\Xendinsertsep@\hskip\csuse{Xendafternote@#4}\csuse{Xendsep@#4}\else\iftoggle{Xendparagraph@#4}{\global\Xendinsertsep@true}{}\fi\xdef\@currentseries{#4}\do##1{\toggletrue{##1@}}\notblank{#5}{\docsvlist{#5}}\csuse{bhookXendnote@#4}\csuse{Xendnotefontsize@#4}{}}\end{verbatim}
The `\setprintendlines` macro is similar to `\printlines` but is for printing endnotes rather than footnotes.

The principal difference between foot- and endnotes is that footnotes are printed on the page where they are specified but endnotes are printed at a different point in the document. We need an indication of the source of an endnote; `\setprintendlines` provides this by always printing the page number. The coding is slightly simpler than `\setprintlines`.

First of all, we print the second page number only if the ending page number is different from the starting page number.

\begin{verbatim}
\setprintendlines  The \texttt{\setprintendlines} macro is similar to \texttt{\printlines} but is for printing endnotes rather than footnotes.

The principal difference between foot- and endnotes is that footnotes are printed on the page where they are specified but endnotes are printed at a different point in the document. We need an indication of the source of an endnote; \texttt{\setprintendlines} provides this by always printing the page number. The coding is slightly simpler than \texttt{\setprintlines}.

First of all, we print the second page number only if the ending page number is different from the starting page number.
\end{verbatim}
We print the ending line number if: (1) we’re printing the ending page number, or (2) it’s different from the starting line number.

We print the starting sub-line if it’s nonzero.

We print the ending sub-line if it’s nonzero and: (1) it’s different from the starting sub-line number, or (2) the ending line number is being printed.
Now we’re ready to print it all.

Now we’re ready to print it all. The only subtlety left here is when to print a period between numbers. But the only instance in which this is tricky is for the ending sub-line number: it could be coming after the starting sub-line number (in which case we want only the dash) or after an ending line number (in which case we need to insert a period).

So, first, print the start lines.

And now, print the dash + the end line number, or the line number range symbol.
\printnpnum A macro to print a page number in an endnote.
\newcommand*{\printnpnum}[1]{p.#1} 
\doendnotes \doendnotes is the command you use to print one series of endnotes; it takes one argument: the series letter of the note series you want to print. \Xendinsertsep@ is set to true at the first note of the series, and to false at the last one.
\newif\ifXendinsertsep@
\newcommand*{\doendnotes}[1]{\l@dend@close
\begingroup
\makeatletter
\expandafter\let\csname #1end\endcsname=\endprint
\input\jobname.end
\global\Xendinsertsep@false%
\endgroup}
\doendnotesbysection \doendnotesbysection is a variant of the previous macro. While \doendnotes print endnotes for all of numbered sections \doendnotesbysection print the endnotes for the first numbered section at its first call for a series, then for the second section at its second call for the same series, then for the third section at its third call for the same series, and so on.
\newcommand*{\doendnotesbysection}[1]{%
\l@dend@close%
\global\expandafter\advance\csname #1end@bysection\endcsname by 1%
\begingroup
\makeatletter
\def\l@d@section##1{%
\ifnumequal{##1}{\csname #1end@bysection\endcsname}%
{\cslet{#1end}{\endprint}}%
{\cslet{#1end}{\@gobblefive}}%
\}%
\input\jobname.end%
\global\Xendinsertsep@false%
\endgroup%
}%
\noendnotes The \noendnotes command is deprecated. You should prefer noend options.
\newcommand*{\noendnotes}{%
\led@war@noendnotesDeprecate%
\global\let\l@dend@stuff=\relax%
\global\chardef\l@d@end=16%
}%
33 Generate series

In this section, X means the name of the series (A, B etc.)

\series The \series macro is an etoolbox list, which contains the name of all series.

\newcommand{\series}{\}

The command \newseries@\series creates a new series of the footnote.

33.1 Test if series is still existing

\xifinlist{#1}{\series}{\led@warn@SeriesStillExist{#1}}%

33.2 Init specific to eledpar

When calling \newseries@ after having loaded eledpar

\ifdefined\newseries@eledpar%
  \newseries@eledpar{#1}%
\fi%

33.3 For critical footnotes

Critical footnotes are those which start with letters. We look for the \nocritical option of eledmac.

33.3.1 Options

\newtoggle{Xparindent@#1}
\newtoggle{Xlemmadisablefontselection@#1}
\csgdef{Xhangindent@#1}{0pt}%
\csgdef{Xragged@#1}{}
\csgdef{hsizetwocol@#1}{0.45 \hsize}%
\csgdef{hsize{#1}}{0.45 \hsize}%
\csgdef{hsize{#1}}{0.3 \hsize}%
33.3.2 Create inserts, needed to add notes in foot

As regards inserts, see chapter 15 of the TeXBook by D. Knuth.
### 33.3 For critical footnotes

#### 33.3.3 Create commands for critical apparatus, \footnote, \latexfootnote etc.

Note the double # in command: it’s because command is made inside another command.

```latex
    \global\notbool{parapparatus@}{\expandafter\newcommand\expandafter *}{\expandafter\newcommand}
    \csname #1footnote\endcsname[2][2]{%
        \ifnum\@edtext@level>0%
            \begingroup%
            \newcommand{\content}{##2}\
            \ifnumberedpar@%
                \ifledRcol%
                    \ifluatex%
                        \footnotelang@lua[\R]%
                    \fi%
                    \@ifundefined{xpg@main@language}{}{ootnotelang@poly[\R]}%
                    \footnoteoptions@[\R]{}{true}%
                    \xright@appenditem%\noexpand\prepare@preXnotes{#1}%
                    \noexpand\prepare@edindex@fornote{\l@d@nums}%
                    \unexpanded{\def\sw@list@inedtext}{\expandafter\unexpanded\expandafter{\sw@inthisedtext}}%
                    \noexpand\csuse{v#1footnote}{#1}{{\l@d@nums}{\expandonce\@tag}{\expandonce\content}}%
                \else%
                    \fi%
                    \@ifundefined{xpg@main@language}{}{ootnotelang@poly}%
                    \footnoteoptions@[\R]{}{true}%
                    \xright@appenditem%\noexpand\prepare@preXnotes{#1}%
                    \noexpand\prepare@edindex@fornote{\l@d@nums}%
                    \unexpanded{\def\sw@list@inedtext}{\expandafter\unexpanded\expandafter{\sw@inthisedtext}}%
                    \noexpand\csuse{v#1footnote}{#1}{{0|0|0|0|0|0|0}{}{##1}}%
                \fi%
            \else%
                \fi%
            \endgroup%
        \fi%
    }

\else%

    \footnotelang@lua%

    \fi%

\else%

    \footnotelang@lua%

    \fi%

\fi%

\@ifundefined{xpg@main@language}{}{ootnotelang@poly}%

\footnoteoptions@[\R]{}{true}%

\xright@appenditem%\noexpand\prepare@preXnotes{#1}%

\noexpand\prepare@edindex@fornote{\l@d@nums}%

\unexpanded{\def\sw@list@inedtext}{\expandafter\unexpanded\expandafter{\sw@inthisedtext}}%

\noexpand\csuse{v#1footnote}{#1}{{0|0|0|0|0|0|0}{}{##1}}%

\fi%

\else%

    \csuse{v#1footnote}{##1}{0}{0}{0}{0}{0}{0}{}{##1}%

\fi%

\endgroup%

\else%

    \endgroup%

\else%

```
We need to be able to modify \eledmac’s footnote macros and restore their defaults.

\section*{33.3.4 Set standard display}

\footnormal{#1}

End of for critical footnotes.

\subsection*{33.4 For familiar footnotes}

Familiar footnotes are those which end with letters. We look for the \nofamiliar option of \eledmac.

\unless\ifnofamiliar@

\subsection*{33.4.1 Options}

\newtoggle{parindentX@#1}
\csgdef{hangindentX@#1}{0pt}\
\csgdef{raggedX@#1}{}\
\csgdef{hsizetwocolX@#1}{.45 \hsize}\
\csgdef{hsizethreecolX@#1}{.3 \hsize}\
\csgdef{colalignX@#1}{\raggedright}\
\csgdef{notenumfontX@#1}{\notenumfont}\
\csgdef{notefontsetupX@#1}{\notefontsetup}\
\csgdef{bhooknoteX@#1}{}\
\csgdef{afterruleX@#1}{0pt}\
\csgdef{beforenotesX@#1}{1.2em \@plus .6em \@minus .6em}\
\csgdef{maxhnotesX@#1}{\ledfootinsdim}\
\newtoggle{notesXwidthliketwocolumns@#1}\

End of for familiar footnotes.

\subsection*{33.4.2 Create tools for familiar footnotes (\footnoteX)}

First, create the \footnoteX command. Note the double \# in command: it is because a command is called inside another command.
33.5 Common options to critical and familiar footnotes

For historical reasons, parafootsep and afternote hooks are common to critical and familiar footnotes.
\csgdef{parafootsep@#1}{\parafootftmsep}%
\csgdef{afternote@#1}{1em plus.4em minus.4em}%

33.6 The endnotes

Endnotes are commands like \Xendnote, where X is a series letter. First, we check for the noend options.
\unless\ifnoend@

33.6.1 The main macro

The \Xendnote macro functions to write one endnote to the .end file. We change \newlinechar so that in the file every space becomes the start of a new line; this generally ensures that a long note doesn’t exceed restrictions on the length of lines in files.
\Xendnote commands called \Xend commands on to the endnote file; these are analogous to the various footfmt commands above, and they take the same arguments. When we process this file, we want to pick out the notes of one series and ignore all the rest. To do that, we equate the end command for the series we want to \endprint, and leave the rest equated to \@gobblefive, which just skips over its five arguments.

\global\cslet{#1end}{\@gobblefive}

We need to store the number of times \doendnotesbysection is called for one series.

\global\expandafter\newcount\csname #1end@bysection\endcsname

33.6.2 The options

\csgdef{Xendtwolines@#1}{}\%
\csgdef{Xendmorethantwolines@#1}{}\%
\newtoggle{Xendtwolinesbutnotmore@#1}{}\%
\newtoggle{Xendtwolinesonlyinsamepage@#1}{}\%
\newtoggle{Xendlemmadisablefontselection@#1}{}\%
\csgdef{Xendnotenumfont@#1}{\notenumfont}\%
\csgdef{Xendnotefontsize@#1}{\notefontsetup}\%
\csgdef{bhookXendnote@#1}{}\%
\csgdef{boxXendlinenum@#1}{0pt}\%
\csgdef{boxXendlinenumalign@#1}{L}\%
\csgdef{boxXendstartlinenum@#1}{0pt}\%
\csgdef{boxXendendlinenum@#1}{0pt}\%
\csgdef{Xendlemmaseparator@#1}{}\%
\csgdef{Xendbeforelemmaseparator@#1}{0em}\%
\csgdef{Xendafterlemmaseparator@#1}{0.5em}\%
\csgdef{Xendinplaceoflemmaseparator@#1}{0.5em}\%
\newtoggle{Xendparagraph@#1}%
\csgdef{Xendafternote@#1}{1em plus.4em minus.4em}\%
\csgdef{Xendsep@#1}{}\%
End of endnotes declaration
\fi%

Dump series in \series
\list\add{\series}{#1}
\}
\% End of \newseries

33.7 Init standards series (A,B,C,D,E,Z)

34 Display

34.1 Change series order
\seriesatbegin \seriesatbegin{(s)} changes the order of series, to put the series \langle s \rangle at the beginning of the list. The series can be the result of a command.
\newcommand{\seriesatbegin}[1]{
\StrDel{\series}{#1}\{\series\}%
\edef\new{}%
\listeadd{\new}{#1}%
\listeadd{\new}{\series}%
\xdef{\series}{\new}%
}
\seriesatend \seriesatend moves the series to the end of the list.
\newcommand{\seriesatend}[1]{
\StrDel{\series}{#1}\{\series\}%
\edef\new{}%
\listeadd{\new}{\series}%
\listeadd{\new}{#1}%
\xdef{\series}{\new}%
}

34.2 Test series order
\ifseriesbefore \ifseriesbefore{\seriesA}{\seriesB}{\true}{\false} expands \true if \seriesA is printed before \seriesB, expands \false otherwise.
\newcommand{\ifseriesbefore}[4]{
\StrPosition{\series}{#1}\{\first\}%
\StrPosition{\series}{#2}\{\second\}%
\ifnum\second<\first\{\false\}%
\ifnum\second>\first\{\true\}%
\false\}
}
34.3 Options

34.3.1 Tools to set options

\texttt{\textbackslash settoggle@series} \texttt{\textbackslash settoggle@series\{\langle series\rangle\}\{\langle toggle\rangle\}\{\langle value\rangle\}} is a generic command to switch toggles for some series. The arguments are:

- \#1 (mandatory): the series for which the hooks should be set. If empty, all the series will be affected.
- \#2 (mandatory): the name of the hook.
- \#3 (mandatory): the new value of toggle (true or false).
- \#4 (optional): if equal to \texttt{reload}, reload the footnote setting (call \texttt{\footnormal} or \texttt{\footparagraph} or ... depending of the footnote display).
- \#5 (optional): if not empty, and if \#1 is empty, change the hook setting for pseudo-series, as \texttt{appref}.

\begin{verbatim}
\newcommandx{\settoggle@series}[5][4,5,usedefault]{% \\
  \def\do##1{% \\
    \global\settoggle{#2@##1}{#3}% \\
    \ifstrequal{#4}{reload}{% \\
      \csuse{foot\csuse{series@display##1}}{##1}% \\
      \csuse{foot\csuse{series@displayX##1}}{##1}% \\
    }{}% \\
  }% \\
  \ifstrempty{#1}{% \\
    \dolistloop{\@series}% \\
    \ifstrempty{#5}{}{% \\
      \docsvlist{#5}% \\
    }% \\
  }{% \\
    \docsvlist{#1}% \\
  }% \\
}\setcommand@series
\end{verbatim}

\texttt{\setcommand@series} \texttt{\setcommand@series\{\langle series\rangle\}\{(command)\}\{\langle value\rangle\}} is a generic command to change hooks into form of commands for some series. The arguments are:

- \#1 (mandatory): the series for which the hooks should be set. If empty, all the series will be affected.
- \#2 (mandatory): the name of the hook.
- \#3 (mandatory): the new value of the hook/command.
- \#4 (optional): if equal to \texttt{reload}, reload the footnote setting (call \texttt{\footnormal} or \texttt{\footparagraph} or ... depending of the footnote display).
34.3 Options

- #5 (optional): if not empty, and if #1 is empty, change the hook setting for pseudo-series, as appref.

\newcommandx{\setcommand@series}[5][4,5,usedefault] {% 
  \def\do##1 { 
    \csedef{#2@##1}{#3} 
    \ifstrequal{#4}{reload}{ 
      \csuse{foot\csuse{series@display##1}}{##1} 
      \csuse{foot\csuse{series@displayX##1}}{##1} 
    }{} 
  } 
  \ifstrempty{#1}{ 
    \dolistloop{\@series} 
    \ifstrempty{#5}{}{ 
      \docsvlist{#5} 
    } 
  }{} 
  \docsvlist{#1} 
}\%

34.3.2 Tools to generate options commands

\newhookcommand@series \newhookcommand@series \command names is a generic command to add new commands for hooks, like \hsizetwocol. The first argument is the name of the hook, the second a comma separated list of pseudo-series where the hook can be used, like appref in the case of \twolines. The second argument is also used to create commands named \hookname<pseudoseries>, like \twolinesappref.

\newcommandx{\newhookcommand@series}[2][2,usedefault] {% 
  \global\expandafter\newcommandx\expandafter*\csname #1\endcsname[2][1,2={true},usedefault]{ 
    \setcommand@series{##1}{#1}{##2} 
  }% 
  \docsvlist{#2} 
}\%

\newhooktoggle@series \newhooktoggle@series \command names is a generic command to add new commands for a new toggle hook, like \numberonlyfirstinline. The second argument is also used to create commands named \hookname<pseudoseries>, like \twolinesbutnotmoreappref.

\newcommandx{\newhooktoggle@series}[2][2,usedefault] {% 
  \global\expandafter\newcommand\expandafter*\csname #1\endcsname[2][2,usedefault]{ 
    \settoggle@series{##1}{#1}{##2} 
  }% 
}\%
\newhooktoggle@series \newhookcommand@toggle@reload does the same thing as \newhooktoggle@series but the commands created by this macro also reload the series which is displayed (normal, paragraph, twocol, threecol).

\newcommand{\newhooktoggle@series@reload}[1]{
  \global\expandafter\newcommand\expandafter*\csname #1\endcsname[2][1,2={true},usedefault]
  \settoggle@series{##1}{#1}{##2}[reload]
}

\newhookcommand@series@reload \newhookcommand@series@reload does the same thing as \newhookcommand@series but the commands created by this macro also reload the series which is displayed (normal, paragraph, twocol, threecol).

\newcommand{\newhookcommand@series@reload}[1]{
  \global\expandafter\newcommand\expandafter*\csname #1\endcsname[2][1,2={true},usedefault]
  \setcommand@series{##1}{#1}{##2}[reload]
}

34.3.3 Options for critical notes

Before generating the commands that are used to set the critical notes, such as \numberonlyfirstineline, \lemmaseparator and the like, we check the nocritical option.

\unless\ifnocritical@
  \newhooktoggle@series{Xparindent}
  \newhookcommand@series{twolines}[appref]
  \newhookcommand@series{trimcolalign}[appref]
  \newhookcommand@series{morethanwolines}[appref]
  \newhookcommand@series{twolinesbutnotmore}[appref]
  \newhooktoggle@series{twolinesonlyinsamepage}[appref]
  \newhookcommand@series{Xhangindent}
  \newhookcommand@series{Xragged}
  \newhookcommand@series{hsizetwocol}
  \newhookcommand@series{hsizethreecol}
  \newhookcommand@series{Xcolalign}\
  \newhookcommand@series{Xnotenumfont}
  \newhookcommand@series{bhookXnote}
  \newhookcommand@series{boxsymlinenum}\
  \newhookcommand@series{symlinenum}
34.3 Options

Before generating the optional commands for familiar notes, we check the \texttt{nofamiliar} option.

\begin{verbatim}
\unless\ifnofamiliar
\newhooktoggle@series{parindentX}
\newhookcommand@series{hangindentX}
\newhookcommand@series{raggedX}
\newhookcommand@series{hsizetwocolX}
\newhookcommand@series{hsizethreecolX}
\newhookcommand@series{colalignX}
\newhookcommand@series{notenumfontX}
\newhookcommand@series{bhooknoteX}
\newhooktoggle@series{beforenotesX}
\newhookcommand@series{maxhnotesX}
\newhooktoggle@series{notesXwidthliketwocolumnsX}
\newhookcommand@series{afterruleX}
\newhookcommand@series{boxlinenumX}
\newhookcommand@series{boxlinenumalignX}
\newhookcommand@series{boxstartlinenumX}
\newhookcommand@series{boxendlinenumX}
\fi

34.3.4 Options for familiar notes

Before generating the optional commands for familiar notes, we check the \texttt{nofamiliar} option.

\begin{verbatim}
\unless\ifnofamiliar
\newhooktoggle@series{parindentX}
\newhookcommand@series{hangindentX}
\newhookcommand@series{raggedX}
\newhookcommand@series{hsizetwocolX}
\newhookcommand@series{hsizethreecolX}
\newhookcommand@series{colalignX}
\newhookcommand@series{notenumfontX}
\newhookcommand@series{bhooknoteX}
\newhooktoggle@series{beforenotesX}
\newhookcommand@series{maxhnotesX}
\newhooktoggle@series{notesXwidthliketwocolumnsX}
\newhookcommand@series{afterruleX}
\end{verbatim}
34.3.5 Common options to critical and familiar footnotes

For historical reasons, \texttt{parafootsep} and \texttt{afternote} hooks are common to critical and familiar footnotes.

\begin{verbatim}
\newhookcommand@series{notefontsizeX}
\fi
\end{verbatim}

34.3.6 Options for endnotes

Before generating the commands that are used to set the endnotes, such as \texttt{numberonlyfirstinline}, \texttt{lemmaseparator} and the like, we check the \texttt{noend} option.

\begin{verbatim}
\unless\ifnoend@
\newhookcommand@series{Xendtwolines}[apprefwithpage]
\newhookcommand@series{Xendmorethantwolines}[apprefwithpage]
\newhooktoggle@series{Xendtwolinesbutnotmore}[apprefwithpage]
\newhooktoggle@series{Xendtwolinesonlyinsamepage}[apprefwithpage]
\newhookcommand@series{Xendnotenumfont}
\newhookcommand@series{bhookXendnote}
\newhookcommand@series{boxXendlinenum}
\newhookcommand@series{boxXendlinenumalign}
\newhookcommand@series{boxXendstartlinenum}
\newhookcommand@series{boxXendendlinenum}
\newhookcommand@series{Xendnotefontsize}
\newhooktoggle@series{Xendlemmadisablefontselection}
\newhookcommand@series{Xendlemmaseparator}
\newhookcommand@series{Xendbeforelemmaseparator}
\newhookcommand@series{Xendafterlemmaseparator}
\newhookcommand@series{Xendinplaceoflemmaseparator}
\newhooktoggle@series{Xendparagraph}
\newhookcommand@series{Xendafternote}
\newhookcommand@series{Xendsep}
\fi
\end{verbatim}

34.4 Old commands, kept for backward compatibility

The next commands are kept for backward compatibility, but should not be used anymore.

\begin{verbatim}
\notenumfont\notefontsetup\newcommand*{\notenumfont}{\normalfont}
\ifledplinenum \newcommand*{\notefontsetup}{\footnotesize}
\symplinenum
\end{verbatim}
34.5 Hooks for a particular footnote

\textbardbl We need to robustify \textbardbl in order to allow it use in \IfStrEq when using as \symplinenum.

35.4.5 Hooks for a particular footnote

\fulllines@ \fulllines@ toggle is used to print the fulllines references, and not the abbreviated form defined by \twolines and \morethantwolines.

\nonum@ \nonum@ toggle is used to disable line number printing in a particular footnote.

\nosep@ \nosep@ toggle is used to disable the lemma separator in a particular footnote.

\nomk@ \nomk@ toggle is used by eledpar to remove the footnote mark in the text when using \footnoteXmk. Read eledpar handbook.

34.6 Alias

\nolemmaseparator \nolemmaseparator[(series)] is just an alias for \lemmaseparator[(series)].

\interparanoteglue The \ipn@skip skip and \interparanoteglue command are kept for backward compatibility, but should not be used anymore.

\parafootftmsep The \parafootftmsep macro is kept for backward compatibility. It is default value of \parafootsep@series.

35 Line number printing

\printlinefootnote The \printlinefootnote macro is called in each \type{footfmt} command. It controls whether the line number is printed or not, according to the previous options. Its first argument is the information about lines; its second argument is the number of lines.
is the series of the footnote. The printing of the line number is shared in \printlinefootnotenumbers.

\newcommand{\printlinefootnote}[2]{%
  \def\extractline@##1|##2|##3|##4|##5|##6|##7|{##2}%
  \def\extractsubline@##1|##2|##3|##4|##5|##6|##7|{##3}%
  \def\extractendline@##1|##2|##3|##4|##5|##6|##7|{##5}%
  \def\extractendsubline@##1|##2|##3|##4|##5|##6|##7|{##6}%
  \iftoggle{numberonlyfirstintwolines@#2}{%
    \edef\lineinfo@{\extractline@ #1| - \extractsubline@ #1| - \extractendline@ #1| - \extractendsubline@ #1|}%
  }{%}
  \edef\lineinfo@{\extractline@ #1| - \extractsubline@ #1|}%
  \iftoggle{numeroonlyfirstinline@#2}{% If for this series the line number must be printed only in the first time.
    \ifcsdef{prevline#2}{% Be sure the \prevline exists.
      \ifcsequal{prevline#2}{\lineinfo@}{% Try it
        \IfStrEq{\csuse{symlinenum@#2}}{}{%
          \hspace{\csuse{inplaceofnumber@#2}}%
        }{%
          \hspace{\csuse{beforesymlinenum@#2}}\csuse{Xnotenumfont@#2}%
          \ifdimequal{\csuse{boxsymlinenum@#2}}{0pt}{}%
          \hspace{\csuse{aftersymlinenum@#2}}%
        }%
        \printlinefootnotearea{#1}{#2}%
      }%
    }{%
      \printlinefootnotearea{#1}{#2}%
    }%
  }{%}
  \printlinefootnotearea{#1}{#2}%
}
\printlinefootnotearea  This macro prints the space before the line number, changes the font, then prints the line number and the space after it. It is called by \printlinefootnote depending of the options about repeating line numbers. The first argument is line information, the second is the notes series (A, B, C. etc.)
\newcommand{\printlinefootnotearea}[2]{%
\printbeforenumberinfootnote[#2]%
\csuse{Xnotenumfont@#2}%
\boxfootnotenumbers[#1]{#2}%
\printafternumberinfootnote[#2]%
}
\boxfootnotenumbers  Depending on the user settings, this macro will box line numbers (or not). The first argument is line information, the second is the notes series (A, B, C. etc.) The previous \printlinefootnotearea calls it.
\newcommand{\boxfootnotenumbers}[2]{%
\ifdimequal{\csuse{boxlinenum@#2}}{0pt}{%
\printlinefootnotenumbers[#1]{#2}%
}{%
\hbox to \csuse{boxlinenum@#2}%
\IfSubStr{RC}{\csuse{boxlinenumalign@#2}}{\hfill}{}%
\printlinefootnotenumbers[#1]{#2}%
\IfSubStr{LC}{\csuse{boxlinenumalign@#2}}{\hfill}{}%
}
\printlinefootnotenumbers  This macro prints, if needed, the pstart number and the line number. The first argument is line information, the second is the notes series (A, B, C. etc.) The previous \boxlinefootnote calls it.
\newcommand{\printlinefootnotenumbers}[2]{%
\xdef@currentseries{#2}%
\ifboolexpr{\booltest{pstartinfootnote@#2} and bool{numberpstart}}{
\printpstart}{}%
\iftoggle{onlypstartinfootnote@#2}{}{\printlines#1|}%
}
\printbeforenumberinfootnote  This macro prints a space (before the line number) in footnote. It is called by \printlinefootnotearea. Its only argument is the series
\newcommand{\printbeforenumberinfootnote}[1]{%}
%  \hspace{\csuse{beforenumberinfootnote@#1}}%
\printafternumberinfootnote  This macro prints the space, adding eventually a \nobreak, after the line number, in footnote. It is called by \printlinefootnotearea. Its only argument is the series
\newcommand{\printafternumberinfootnote}[1]{%}
%  \iftoggle{nonbreakableafternumber@#1}{\nobreak}{}%  \hspace{\csuse{afternumberinfootnote@#1}}%

36  Output routine

Now we begin the output routine and associated things.
\pageno  \pageno is a page number, starting at 1, and \advancepageno increments the number.
\countdef\pageno=0 \pageno=1
\newcommand*{\advancepageno}{\ifnum\pageno<\z@ \global\advance\pageno\m@ne
%  \else\global\advance\pageno\@ne\fi}

The next portion is probably the trickiest part of moving from TeX to LaTeX. The original code is below, but we need something very different. This is a new output routine, with changes to handle printing all our footnotes. Those changes have not been added directly, but are in macros that get called here: that should make it easier to see what would need to be taken over to a different output routine. We continue to use the \pagebody, \makeheadline, \makefootline, and \dosupereject macros of Plain TeX; for those macros, and the original version of \output, see The TeXbook, p. 364.

\output{\edmac@output}
\def\edmac@output{\shipout\vbox{\normal@pars
  \vbox{\makeheadline\pagebody\makefootline}%
}%
\advancepageno
%  \iftoggle{outputpenalty}>-\@MM\else\dosupereject\fi}
\def\pagecontents{\page@start
%  \ifvoid\topins\else\unvbox\topins\fi
%  \dimen@=\dp\@cclv \unvbox\@cclv \% open up \box255
%  \do@feet
%  \ifrggedbottom \kern-\dimen@ \vfil \fi}
\texttt{\do@feet} ships out all the footnotes. Standard \texttt{EDMAC} has only five feet, but there is nothing in principal to prevent you from creating an arachnoid or centipede edition; straightforward modifications of \texttt{EDMAC} are all that’s required. However, the myriapedal edition is ruled out by \texttt{eTeX} limitations: the number of insertion classes is limited to $2^{16}$.

With luck we might only have to change \texttt{@makecol} and \texttt{@reinserts}. The kernel definition of these, and perhaps some other things, is:

\begin{verbatim}
\gdef \@makecol {\%
  \ifvoid\footins
    \setbox\@outputbox \box\@cclv
  \else
    \setbox\@outputbox \vbox {\%
      \boxmaxdepth \@maxdepth
      \@tempdima\dp\@cclv
      \unvbox \@cclv
      \vskip \skip\footins
      \color@begingroup
      \normalcolor
      \footnoterule
      \unvbox \footins
      \color@endgroup
    }\%
  \fi
  \xdef\@freelist{\@freelist\@midlist}\%
  \global \let \@midlist \@empty
  \@combinefloats
  \ifvbox\@kludgeins
    \@makespecialcolbox
  \else
    \setbox\@outputbox \vbox to\@colht {\%
      \@texttop
      \dimen@ \dp\@outputbox
      \unvbox\@outputbox
      \vskip \dimen@
      \@textbottom
    }\%
  \fi
  \global \maxdepth \@maxdepth
}\%
\gdef \@reinserts{\%
  \ifvoid\footins \else \insert\footins{\unvbox\footins}\fi
  \ifvbox\@kludgeins \insert\@kludgeins{\unvbox\@kludgeins}\fi
}\%

\end{verbatim}

Now we start actually changing things.
\@makecol.
\providecommand{\m@m@makecolfloats}{% 
\xdef\@freelist{\@freelist\@midlist}% 
\global\let\@midlist\@empty 
\@combinefloats}
\providecommand{\m@m@makecoltext}{% 
\ifvbox\@kludgeins 
\@makespecialcolbox 
\else 
\setbox\@outputbox\vbox to@colht {% 
\@texttop 
\dimen@\dp\@outputbox 
\unvbox\@outputbox 
\vskip -\dimen@ 
\@textbottom}% 
\fi} 
\providecommand{\m@m@makecolintro}{% 
\l@d@makecol

This is a partitioned version of the 'standard' \@makecol, with the initial code put into another macro.
\gdef\l@d@makecol{% 
\l@ddofootinsert 
\m@m@makecolfloats 
\m@m@makecoltext 
\global\maxdepth\@maxdepth} 
\ifFN@bottom The \ifFN@bottom macro is defined by the footmisc package. If this package is not loaded, we define it.
\AtBeginDocument{%\ifpackageloaded{footmisc}{\newif\ifFN@bottom} 
\l@ddofootinsert

This macro essentially holds the initial portion of the kernel \@makecol code.
\newcommand*{%\l@ddofootinsert}{}
\ifvoid\footins 
\setbox\@outputbox\vbox\box\@cclv 
\else 
\setbox\@outputbox\vbox {% 
\boxmaxdepth\@maxdepth 
\@tempdima\dp\@cclv 
\unvbox\@cclv 
\ifFN@bottom\vfill\fi\vskip\skip\footins\% If the option bottom of loadmisc package is loaded
\color@begingroup 
\normalcolor 
\footnoterule 
\unvbox\footins 
\color@endgroup 
}%
That's the end of the copy of the kernel code. We finally call a macro to handle all the additional EDMAC feet.

\doxtrafeet
\doxtrafeet is the code extending \@makecol to cater for the extra eledmac feet. We have two classes of extra footnotes. By default, we order the footnote inserts so that the regular footnotes are first, then class 1 (familiar footnotes) and finally class 2 (critical footnotes).

\newcommand*{\l@ddoxtrafeet}{%
  \IfStrEq{familiar-critical}{\@fnpos}{\doxtrafeeti\doxtrafeetii}%
  \IfStrEq{critical-familiar}{\@fnpos}{\doxtrafeetii\doxtrafeeti}%
  \doxtrafeeti\doxtrafeetii}%
\doxtrafeetii
\doxtrafeetii is the code extending \@makecol to cater for the extra critical feet (class 2 feet). NOTE: the code is likely to be ‘featurefull’.

\newcommand*{\doxtrafeetii}{%
  \setbox\@outputbox \vbox{\unvbox\@outputbox \@opxtrafeetii}}
\@opxtrafeetii
\@opxtrafeetii is the extra critical feet to be added to the output. The normal way to add one series. \print@Xnotes is replaced by eledpar when using \Pages.

\newcommand{\print@Xnotes}[1]{%
  \csuse{#1footstart}{#1}%
  \csuse{#1footgroup}{#1}%%
}\@xnotes
We print all series of notes by looping on them. We check before printing them that they are not voided.

\newcommand*{\@opxtrafeetii}{%
  \unless\ifnocritical0%
  \gdef\firstXseries{0}%
  \do\if#1%
    \ifvoid\csuse{#1footins}%
      \global\skip\csuse{#1footins}=\csuse{beforeXnotes}@#1%
      \global\advance\skip\csuse{#1footins} by\csuse{afterXrule}@#1%
      \print@Xnotes{#1%
    \fi%
  }%
  \dolistloop{\@series}%
\l@ddodoreinxtrafeet \l@ddodoreinxtrafeet is the code for catering for the extra footnotes within \@reinserts. The implementation may well have to change. We use the same classes and ordering as in \l@ddoxtrafeet.

\newcommand*{\l@ddodoreinxtrafeet}{\doreinxtrafeeti\doreinxtrafeetii}
\newcommand*{\doreinxtrafeetii}{\unless\ifnocritical@\%
\def\do##1{\ifvoid\csuse{##1footins}\else\insert\csuse{##1footins}{\unvbox\csuse{##1footins}}\fi}\
\dolistloop{\@series}\fi\%
\dolistloop{\@series}
\fi\%
}\l@d@reinserts
And here is the modified version of \@reinserts.
\gdef \l@d@reinserts{\l@d@makecol\l@d@reinserts\fi}
\ifvoid\footins\else\insert\footins{\unvbox\footins}\fi
\l@ddodoreinxtrafeet\l@ddodoreinxtrafeet
\ifvbox\@kludgeins\insert\@kludgeins{\unvbox\@kludgeins}\fi
}\l@d@makecol
\gdef \@reinserts{\l@d@reinserts\fi}

The memoir class does not use the ‘standard’ versions of \@makecol and \@reinserts, due to its sidebar insert. We had better add that code if memoir is used. (It can be awkward dealing with \if code within \if code, so don’t use \ifl@dmemoir here.)
\gdef \l@d@reinserts{%
\g@addto@macro{\m@mdoextrafeet}{\l@ddoxtrafeet}%
\g@addto@macro{\m@mdodoreinextrafeet}{\l@ddodoreinxtrafeet}%
}%
\l@d@reinserts
\l@d@reinserts
\g@addto@macro{\m@mdoextrafeet}{\l@ddoxtrafeet}\
\g@addto@macro{\m@mdodoreinextrafeet}{\l@ddodoreinxtrafeet}\
\l@d@reinserts
\l@d@reinserts

\addfootins \addfootins is for backward compatibility, but shouldn't be used anymore.

\newcommand*{\addfootins}[1]{%
  \led@warn@AddfootinsObsolete%
  \footnormal{#1}
  \ifvoid\nameuse{#1footins}else
    \nameuse{#1footstart}{#1}\nameuse{#1footgroup}{#1}\fi
  \g@addto@macro{\@opxtrafeetii}{%
    \ifvoid\@nameuse{#1footins}else
      \@nameuse{#1footins}\@nameuse{#1footgroup}{#1}\fi}%
  \g@addto@macro{\doreinxtrafeetii}{%
    \ifvoid\@nameuse{#1footins}else
      \insert\@nameuse{#1footins}\unvbox\@nameuse{#1footins}\fi}
  \g@addto@macro{\l@dedbeginmini}{%
    \expandafter\let\csname #1footnote\endcsname = \@nameuse{mp#1footnote}}
  \g@addto@macro{\l@dedendmini}{%
    \ifvoid\@nameuse{mp#1footins}else\@nameuse{mpfootgroup#1}{#1}\fi}
}%

It turns out that \addclearpage also needs modifying.

\if@led@nofoot
  We have to check if there are any leftover feet. \add@extranote is a hook for handling further footnotes.
\fi
\newif\if@led@nofoot
\newcommand*{\@led@extranote}{{}%}
\ifclassloaded{memoir}{%
  If the \texttt{memoir} class is loaded we hook into its modified \addclearpage.
\g@addto@macro{\@mem@extranote}{%
  \unless\ifnocritical%
    \do#1{\ifvoid\csuse{#1footins}else\@mem@nofootfalse\fi%
      \ifvoid\csuse{footins#1}false\fi%}
  \dolistloop{\@series}%
  \if@led@extranote%}
  \fi%}
\newcommand*{\@led@extranote}{%}
\if@led@nofoot%
  %
\fi%
%}

As \texttt{memoir} is not loaded we have to do it all here.

\if@led@testifnofoot
\newcommand*{\@led@testifnofoot}{%}
\ifvoid\footins\else\@led@nofoottrue\fi%
\def\do##1{%}
\unless\ifnocritical%
  \ifvoid\csuse{##1footins}false\fi%}
\dolistloop{\@series}%
\@led@extranote%
\fi%}
\fi%
%
\if@led@nofootfalse\fi\fi
\unless\ifnofamiliar\fi\fi
\ifvoid\csuse{footins##1}\else\@led@nofootfalse\fi\fi
}\dolistloop{\@series}\@led@extranofeet
\renewcommand{\@doclearpage}{\@led@testifnofoot\if@led@nofoot\setbox\@tempboxa\vsplit\@cclv to\z@ \unvbox\@tempboxa\setbox\@tempboxa\box\@cclv\xdef\@deferlist{\@toplist\@botlist\@deferlist}\global \let\@toplist\@empty\global \let\@botlist\@empty\global \@colroom \@colht\ifx \@currlist\@empty\else\@latexerr{Float(s) lost}@ehb\global \let\@currlist\@empty\fi\@makefcolumn\@deferlist\@whilesw\if@fcolmade \fi{\@opcol\@makefcolumn\@deferlist}\if@twocolumn\if@firstcolumn\xdef\@dbldeferlist{\@dbltoplist\@dbldeferlist}\global \let\@dbltoplist\@empty\global \@colht \textheight\begingroup\@dblfloatplacement\@makefcolumn\@dbldeferlist\@whilesw\if@fcolmade \fi{\@outputpage\@makefcolumn\@dbldeferlist}\endgroup\else\vbox{}\clearpage\fi\fi\@makefcolumn\@deferlist\@whilesw\if@fcolmade \fi{\@opcol\@makefcolumn\@deferlist}\if@twocolumn\if@firstcolumn\xdef\@dbldeferlist{\@dbltoplist\@dbldeferlist}\global \let\@dbltoplist\@empty\global \@colht \textheight\begingroup\@dblfloatplacement\@makefcolumn\@dbldeferlist\@whilesw\if@fcolmade \fi{\@outputpage\@makefcolumn\@dbldeferlist}\endgroup\else\vbox{}\clearpage\fi\fi\else\setbox\@cclv\vbox{\box\@cclv\vfil}\l@d@makecol\@opcol\clearpage\fi}
37 Cross referencing

Peter Wilson has rewritten portions of the code in this section so that the LaTeX .aux file is used. This will also handle \include files.

Further, I have renamed some of the original EDMAC macros so that they do not clash with the LaTeX label/ref commands (EDMAC and LaTeX use very different mechanisms). In particular, the original EDMAC \label and \pageref have been renamed as \edlabel and \edpageref respectively.

You can mark a place in the text using a command of the form \edlabel{foo}, and later refer to it using the label foo by saying \edpageref{foo}, or \lineref{foo} or \sublineref{foo}. These reference commands will produce, respectively, the page, line and sub-line on which the \edlabel{foo} command occurred.

The reference macros warn you if a reference is made to an undefined label. If foo has been used as a label before, the \edlabel{foo} command will issue a complaint; subsequent \edpageref and \edlineref commands will refer to the latest occurrence of \label{foo}.

\labelref@list Set up a new list, \labelref@list, to hold the page, line and sub-line numbers for each label.

3881 \list@create{\labelref@list}

\zz@@@ A convenience macro to zero two labeling counters in one go.
3882 \% \newcommand*{\zz@@@}{000|000} % set three counters to zero in one go
3883 \newcommand*{\zz@@@}{000|000} % set two counters to zero in one go
3884
\edlabel The \edlabel command first writes a \@lab macro to the \linenum@out file. It then checks to see that the \labelref@list actually has something in it (if not, it creates a dummy entry), and pops the next value for the current label, storing it in \label@refs. Finally it defines the label to be \empty so that any future check will turn up the fact that it has been used.31

This version of the original EDMAC \label uses \@bsphack and \@esphack to eliminate extra space problems and also the LaTeX write methods for the .aux file.

Jesse Billett34 found that the original code could be off by several pages. This version, hopefully cures that, and also allows for non-arabic page numbering.

3885 \newcommand*{\edlabel}[1]{% 
3886 \ifl@dpairing\ifautopar% 
3887 \strut% 
3888 \fi\fi%

31 The remaining macros in this section were kindly revised by Wayne Sullivan, who substantially improved their efficiency and flexibility.
34 [db430@cam.ac.uk] via the ctt thread ‘ledmac cross referencing’, 25 August 2003.
Cross referencing

Use code from the kernel `\label` command to write the correct page number (it seems possible that the original EDMAC's `\page@num` scheme might also have had problems in this area). Also define an hypertarget if `hyperref` package is loaded.

```
\protected@write\@auxout{}\% 
\ifdef\hypertarget\Hy@raisedlink\hypertarget{#1}{}}\% 
\else\% 
\write\linenum@out\% 
\ifx\labelref@list\empty\% 
\xdef\label@refs{\zz@@@}\% 
\else\% 
\gl@p\labelref@list\to\label@refs\% 
\fi\% 
\ifvmode\% 
\advancelabel@refs\% 
\fi\% 
\protected@write\@auxout{}\% 
\ifdef\hypertarget\Hy@raisedlink\hypertarget{#1}{}}\% 
\else\% 
\write\linenum@out\% 
\ifx\labelref@list\empty\% 
\xdef\label@refs{\zz@@@}\% 
\else\% 
\gl@p\labelref@list\to\label@refs\% 
\fi\% 
\ifvmode\% 
\advancelabel@refs\% 
\fi\% 
\protected@write\@auxout{}\% 
\ifdef\hypertarget\Hy@raisedlink\hypertarget{#1}{}}\% 
\else\% 
\write\linenum@out\% 
\ifx\labelref@list\empty\% 
\xdef\label@refs{\zz@@@}\% 
\else\% 
\gl@p\labelref@list\to\label@refs\% 
\fi\% 
\ifvmode\% 
\advancelabel@refs\% 
\fi\% 
\protected@write\@auxout{}\% 
\ifdef\hypertarget\Hy@raisedlink\hypertarget{#1}{}}\% 
```

In cases where `\edlabel` is the first element in a paragraph, we have a problem with line counts, because line counts change only at the first horizontal box of the paragraph. Hence, we need to test `\edlabel` if it occurs at the start of a paragraph. To do so, we use `\ifvmode`. If the test is true, we must advance by one unit the amount of text we write into the `.aux` file. We do so using `\advancelabel@refs` command.

```
\newcounter{line}\% 
\newcounter{subline}\% 
\newcommand{\advancelabel@refs}\% 
\setcounter{line}{\expandafter\labelrefsparseline\label@refs}\% 
\stepcounter{line}\% 
\ifsublines\% 
\setcounter{subline}{\expandafter\labelrefsparsesubline\label@refs}\% 
\stepcounter{subline}{1}\% 
```
\l@dmake@labels  The \l@dmake@labels macro gets executed when the labels file is read. For each label it defines a macro, whose name is made up partly from the label you supplied, that contains the page, line and sub-line numbers. But first it checks to see whether the label has already been used (and complains if it has).

The initial use of \newcommand is to catch if \l@dmake@labels has been previously defined (by a class or package).

\newcommand*{\l@dmake@labels}{
\expandafter\ifx\csname the@label#5\endcsname \relax\else\led@warn@DuplicateLabel{#5}\fi
\expandafter\gdef\csname the@label#5\endcsname{#1|#2|#3|#4}\
\ignorespaces}

LaTeX reads the aux file at both the beginning and end of the document, so we have to switch off duplicate label checking after the first time the file is read.

\AtBeginDocument{\def\l@dmake@labels#1|#2|#3|#4|#5{}}

\applabel, if called in \edtext will insert automatically both a start and an end label for the current edtext lines.

\newcommand*{\applabel}{\xright@appenditem
{\linenumrep{\line@num}|\ifsublines@ \sublinenumrep{\subline@num}\else 0\fi}\to\labelref@list}
Label should not be already defined.

```
\ifcsundef{the@label#1}{% 
  \csdef{the@label#1}{applabel}% 
} {% 
  \led@warn@DuplicateLabel{#1 (applabel)}% 
}
```

Parse the edtext line numbers.

```
\expandafter\l@dp@rsefootspec\l@d@nums|%
```

Use the \LaTeXstandard hack for label.

```
\@bsphack%
```

And now, write the data in the auxiliary file.

```
\ifledRcol1% 
  \protected@write\@auxout{}{% 
    {\string\l@dmake@labelsR\space\l@dparsedstartpage|\l@dparsedstartline|\l@dparsedstartsub|\the\c@pstartR|{#1:start}}% 
    \ifdef\hypertarget\Hy@raisedlink{\hypertarget{#1:start}{}}{}% 
    \protected@write\@auxout{}{% 
      {\string\l@dmake@labelsR\space\l@dparsedendpage|\l@dparsedendline|\l@dparsedendsub|\the\c@pstartR|{#1:end}}% 
    }% 
  }% 
\else% 
  \protected@write\@auxout{}{% 
    {\string\l@dmake@labels\space\l@dparsedstartpage|\l@dparsedstartline|\l@dparsedstartsub|\the\c@pstart|{#1:start}}% 
    \ifdef\hypertarget\Hy@raisedlink{\hypertarget{#1:start}{}}{}% 
    \protected@write\@auxout{}{% 
      {\string\l@dmake@labels\space\l@dparsedendpage|\l@dparsedendline|\l@dparsedendsub|\the\c@pstart|{#1:end}}% 
    }% 
  }% 
\fi% 
```

Use the \LaTeXstandard hack for label.

```
\@esphack%
```

Warning if \edlabel is called outside of edtext.

```
\else% 
  \led@warn@AppLabelOutEdtext{#1}% 
\fi% 
End of \applabel
```

\wrap@edcrossref is called around all edlemac crossref commands, except those which start with x. It adds the hyperlink.

```
\newrobustcmd{\wrap@edcrossref}[2]{% 
  \ifdef\hyperlink\{\hyperlink{#1}{#2}\}{}% 
  \{#2\} %
}
```

\edpageref If the specified label exists, \edpageref gives its page number. For this reference command, as for the other two, a special version with prefix x is provided for use in places where the command is to be scanned as a number, as in \\linenum.
These special versions have two limitations: they don’t print error messages if
the reference is unknown, and they can’t appear as the first label or reference
command in the file; you must ensure that a \texttt{edlabel} or a normal reference
command appears first, or these \texttt{x}-commands will always return zeros. \LaTeX
already defines a \texttt{pageref}, so changing the name to \texttt{edpageref}.

\pagebreak

\texttt{edpageref} \textbf{If} the specified label exists, \texttt{lineref} gives its line number.
\texttt{\textbackslash lineref} 3986 \texttt{newcommand*{\textbackslash edlineref}[1]{\textbackslash \l@dref@undefined{#1}\wrap@edcrossref{#1}{\textbackslash \l@dgetref@num{1}{#1}}}}\newcommand*{\textbackslash xlineref}[1]{\textbackslash \l@dgetref@num{1}{#1}}

\texttt{\textbackslash xlineref} 3987 \AtBeginDocument{\ifdef\lineref{}{\let\lineref\edlineref}{}\textbackslash xlineref} 3988 \texttt{newcommand*{\textbackslash xlineref}[1]{\textbackslash \l@dgetref@num{2}{#1}}}\newcommand*{\xlineref}[1]{\textbackslash \l@dgetref@num{2}{#1}}

\texttt{\textbackslash sublineref} \textbf{If} the specified label exists, \texttt{sublineref} gives its sub-line number.
\texttt{\textbackslash xsublineref} 3992 \texttt{newcommand*{\textbackslash sublineref}[1]{\textbackslash \l@dref@undefined{#1}\wrap@edcrossref{#1}{\textbackslash \l@dgetref@num{3}{#1}}}}\newcommand*{\textbackslash xsublineref}[1]{\textbackslash \l@dgetref@num{3}{#1}}

\texttt{\textbackslash pstartref} \textbf{If} the specified label exists, \texttt{pstartref} gives its pstart number.
\texttt{\textbackslash xpstartref} 3995 \texttt{newcommand*{\textbackslash pstartref}[1]{\textbackslash \l@dref@undefined{#1}\wrap@edcrossref{#1}{\textbackslash \l@dgetref@num{4}{#1}}}}\newcommand*{\textbackslash xpstartref}[1]{\textbackslash \l@dgetref@num{4}{#1}}

The next three macros are used by the referencing commands above, and do the
job of extracting the right numbers from the label macro that contains the page,
line, and sub-line number.

\texttt{\textbackslash \l@dref@undefined} The \texttt{\textbackslash \l@dref@undefined} macro is called when you refer to a label with the normal
referencing macros. Its argument is a label, and it just checks that the label has
been defined.
\texttt{\textbackslash \l@dref@undefined} 3998 \texttt{newcommand*{\textbackslash \l@dref@undefined}[1]{\%}}\expandafter\ifx\csname the@label#1\endcsname\relax\texttt{\led@warn@RefUndefined{#1}{\%}}\fi\expandafter\ifx\csname the@label#1\endcsname\relax

\texttt{\textbackslash \l@dgetref@num} Next, \texttt{\textbackslash \l@dgetref@num} fetches the number we want. It has two arguments: the
first is simply a digit, specifying whether to fetch a page (1), line (2) or sub-line
(3) number. (This switching is done by calling \texttt{\textbackslash \l@dlabel@parse}.) The second
argument is the label-macro, which because of the \texttt{\textbackslash \l@lab} macro above is defined
to be a string of the type 123|456|789.
\texttt{\textbackslash \l@dgetref@num} 4003 \texttt{newcommand*{\textbackslash \l@dgetref@num}[2]{\%}}\expandafter\ifx\csname the@label#1\endcsname\relax

\l@dl@label@parse Notice that we slipped another | delimiter into the penultimate line of \l@dgetref@num, to keep the ‘switch-number’ separate from the reference numbers. This | is used as another parameter delimiter by \l@dl@label@parse, which extracts the appropriate number from its first arguments. The |-delimited arguments consist of the expanded label-macro (three reference numbers), followed by the switch-number (1, 2, or 3) which defines which of the earlier three numbers to pick out. (It was earlier given as the first argument of \l@dgetref@num.)

\xxref The \xxref command takes two arguments, both of which are labels, e.g., \xxref{mouse}{elephant}. It first does some checking to make sure that the labels do exist (if one doesn’t, those numbers are set to zero). Then it calls \l@linenum and sets the beginning page, line, and sub-line numbers to those of the place where \l@label{mouse} was placed, and the ending numbers to those at \l@label{elephant}. The point of this is to be able to manufacture footnote line references to passages which can’t be specified in the normal way as the first argument to \l@crite@xt for one reason or another. Using \xxref in the second argument of \l@crite@xt lets you set things up at least semi-automatically.

\xxref The \xxref command takes two arguments, both of which are labels, e.g., \xxref{mouse}{elephant}. It first does some checking to make sure that the labels do exist (if one doesn’t, those numbers are set to zero). Then it calls \l@linenum and sets the beginning page, line, and sub-line numbers to those of the place where \l@label{mouse} was placed, and the ending numbers to those at \l@label{elephant}. The point of this is to be able to manufacture footnote line references to passages which can’t be specified in the normal way as the first argument to \l@crite@xt for one reason or another. Using \xxref in the second argument of \l@crite@xt lets you set things up at least semi-automatically.
\letcs{\@tempa}{the@@label#1}\
\letcs{\@tempb}{the@@label#2}\
\fi\
\linenum{\@tempa|\@tempb}}\

\appref\appprefwithpage\appprefprefixsingle\appprefprefixmore

\appref prints a crossref to some lines of the apparatus defined by \applabel. It
prints the lines as they should be printed in the apparatus.

If \appprefprefixsingle is not empty, it prints it before the line number. If
\appprefprefixsingles is not empty, it prints it before the line numbers when
the first line is not the same as the last line. \appprefwithpage prints a cross-
ref to some lines of the apparatus defined by \applabel. It always prints the
page number, as it should be printed in the end notes. The \twolinesappref
and \morethantwolinesappref are similar to the footnote hooks and \twolines
\morethantwolines.

So, first declare the default value of the hooks for the pseudo-series appref.
Also declare the internal toggle which are switch by eledmac.

\xdef\twolines@appref{}\
\xdef\morethantwolines@appref{}\
\newtoggle{twolinesbutnotmore@appref}\
\newtoggle{twolinesonlyinsamepage@appref}\
\newtoggle{\Xendtwolines@apprefwithpage}\
\newtoggle{\Xendmorethantwolines@apprefwithpage}\
\newtoggle{\Xendtwolinesbutnotmore@apprefwithpage}\
\newtoggle{\Xendtwolinesonlyinsamepage@apprefwithpage}\

Note that some of these hooks are declared but no user command can change their
values. Such hooks are not pertinent for appref and apprefwithpage pseudo-series,
but their values are nonetheless tested in some macros.

\xdef\boxstartlinenum@appref{0pt}\
\xdef\boxendlinenum@appref{0pt}\
\xdef\boxXendstartlinenum@apprefwithpage{0pt}\
\xdef\boxXendendlinenum@apprefwithpage{0pt}\

Now, declare the default value of \appprefprefixsingle and \appprefprefixmore.
\newcommand{\appprefprefixsingle}{0pt}\
\newcommand{\appprefprefixmore}{0pt}\

And now, the main commands: \appref and \appprefwithpage. These commands
call \printlines and \printendlines. That is why we have previously declared
all hooks values tested inside these last commands.
\newcommandx{\appref}{2}[1,usedefault]{%\
\IfStrEq{#1}{fulllines}{%
Sometimes the \texttt{\label} command cannot be used to specify exactly the page and line desired; you can use the \texttt{\makelabel} macro to make your own label. For example, if you say \texttt{\makelabel{elephant}{10|25|0}} you will have created a new label, and a later call to \texttt{\pageref{elephant}} would print ‘10’ and \texttt{\lineref{elephant}} would print ‘25’. The sub-line number here is zero. \texttt{\makelabel} takes a label, followed by a page and a line number(s) as arguments. LaTeX defines a \texttt{\makelabel} macro which is used in lists. I’ve changed the name to \texttt{\makelabel}.

\begin{verbatim}
\newcommand*{\makelabel}[2]{\expandafter\xdef\csname the@label#1\endcsname{#2}}
\end{verbatim}

(If you are only going to refer to such a label using \texttt{\xref}, then you can omit entries in the same way as with \texttt{\linenum} (see \texttt{22.3 p. 97} and \texttt{21.3 p. 72}), since \texttt{\xref} makes a call to \texttt{\linenum} in order to do its work.)

### 38 Side notes

Regular \texttt{\marginpar}s do not work inside numbered text — they don’t produce any note but do put an extra unnumbered blank line into the text.

Changing \texttt{\xympar} a little at least ensures that \texttt{\marginpar}s in numbered text do not disturb the flow.
We provide side notes as replacement for `\marginpar` in numbered text.

These are the sidenote equivalents to `\linemargin` and `\linenummargin` for specifying which margin. The default is the right margin (opposite to the default for line numbers). `\ldgetsidenote@margin` returns the number associated to side note margin:

- left: 0
- right: 1
- outer: 2
- inner: 3
\@dlp@rbox We need two boxes to store sidenote texts.
\@drp@rbox
\newbox\@dlp@rbox
\newbox\@drp@rbox
\ledlsnotewidth
\ledrsnotewidth
\ledlsnotesep
\ledrsnotesep
\ledlsnotefontsetup
\ledrsnotefontsetup
\ledleftnote \ledrightnote \ledinnernote \ledouternote are the user commands for left, right, inner and outer sidenotes. The two last one are just alias for the two first one, depending of the page number. \ledsidenote{⟨text⟩}
\ledsidenote is the command for a moveable sidenote.
\newcommand*[⟨\ledleftnote⟩][1]{\edtext{}\l@dlsnote{#1}}
\newcommand*[⟨\ledrightnote⟩][1]{\edtext{}\l@drsnote{#1}}
\newcommand*[⟨\ledinnernote⟩][1]{\ifodd\c@page% Do not use \page@num, because it is not yet calculated when command is called
\ledleftnote{#1}%
\else%
\ledrightnote{#1}%
\fi%}
\newcommand*[⟨\ledouternote⟩][1]{\ifodd\c@page% Do not use \page@num, because it is not yet calculated when command is called
\ledrightnote{#1}%
\else%
\ledleftnote{#1}%
\fi%}
\newcommand*[⟨\ledsidenote⟩][1]{\edtext{}\l@dcsnote{#1}}
The ‘footnotes’ for left, right, and moveable sidenotes. The whole scheme is reminiscent of the critical footnotes code.
\newif\rightnoteup
\rightnoteuptrue
\newcommand*{\l@dlnote}[1]{\begingroup
\newcommand{\content}{#1}
\ifnumberedpar\ifledRcol
\xright@appenditem{\noexpand\vl@dlnote{\expandonce\content}}
\to\inserts@listR
\global\advance\insert@countR \@ne\else
\xright@appenditem{\noexpand\vl@dlnote{\expandonce\content}}
\to\inserts@list
\global\advance\insert@count \@ne\fi\fi\ignorespaces\endgroup}
\newcommand*{\l@drsnote}[1]{\begingroup
\newcommand{\content}{#1}
\ifnumberedpar\ifledRcol
\xright@appenditem{\noexpand\vl@drsnote{\expandonce\content}}
\to\inserts@listR
\global\advance\insert@countR \@ne\else
\xright@appenditem{\noexpand\vl@drsnote{\expandonce\content}}
\to\inserts@list
\global\advance\insert@count \@ne\fi\fi\ignorespaces\endgroup}
\newcommand*{\l@dcsnote}[1]{\begingroup
\newcommand{\content}{#1}
\ifnumberedpar\ifledRcol
\xright@appenditem{\noexpand\vl@dcsnote{\expandonce\content}}
\to\inserts@listR
\global\advance\insert@countR \@ne\else
\xright@appenditem{\noexpand\vl@dcsnote{\expandonce\content}}
\to\inserts@list
\global\advance\insert@count \@ne\fi\fi\ignorespaces\endgroup}
Put the left/right text into boxes, but just save the moveable text. `\l@dcspotext`, `\l@dcnote@l` and `\l@dcnote@r` are etoolbox lists which will store the content of side notes. We store the content in lists, because we need to loop later on them, in case many sidenote co-exist for the same line. That is there some special test to do, in order to:

- Store the content of `\ledsidenote` to `\l@dcnoteext` in any cases.
- Store the content of `\right sidenote` to:
  - `\l@dcnoteext` if `\ledsidenote` is to be put on right.
  - `\l@dcnoteext@r` if `\ledsidenote` is to be put on left.
- Store the content of `\left sidenote` to:
  - `\l@dcnoteext` if `\ledsidenote` is to be put on left.
  - `\l@dcnoteext@l` if `\ledsidenote` is to be put on right.

```latex
\newcommand*{\vl@dlsnote}[1]{% 
  \ifledRcol@% 
    \@l@dtempcntb=\sidenote@marginR% 
    \ifnum\@l@dtempcntb>\@ne% 
    \advance\@l@dtempcntb by\page@numR% 
    \fi% 
  \else% 
    \@l@dtempcntb=\sidenote@margin% 
    \ifnum\@l@dtempcntb>\@ne% 
    \advance\@l@dtempcntb by\page@num% 
    \fi% 
  \fi% 
  \ifodd\@l@dtempcntb% 
    \listgadd{\l@dcsnotetext@l}{#1}% 
  \else% 
    \listgadd{\l@dcsnotetext}{#1}% 
  \fi% 
}% 
\newcommand*{\vl@drsnote}[1]{% 
  \ifledRcol@% 
    \@l@dtempcntb=\sidenote@marginR% 
    \ifnum\@l@dtempcntb>\@ne% 
    \advance\@l@dtempcntb by\page@numR% 
    \fi% 
  \else% 
    \@l@dtempcntb=\sidenote@margin% 
    \ifnum\@l@dtempcntb>\@ne% 
    \advance\@l@dtempcntb by\page@num% 
    \fi% 
  \fi% 
  \ifodd\@l@dtempcntb% 
    \listgadd{\l@dcsnotetext@l}{#1}% 
  \else% 
    \listgadd{\l@dcsnotetext}{#1}% 
  \fi% 
}% 
\newcommand*{\vl@dcsnote}{% 
  \ifledRcol@% 
    \@l@dtempcntb=\sidenote@marginR% 
    \ifnum\@l@dtempcntb>\@ne% 
    \advance\@l@dtempcntb by\page@numR% 
    \fi% 
  \else% 
    \@l@dtempcntb=\sidenote@margin% 
    \ifnum\@l@dtempcntb>\@ne% 
    \advance\@l@dtempcntb by\page@num% 
    \fi% 
  \fi% 
  %} 
```
This macro is used to separate sidenotes of the same line.
\newcommand{\sidenotesep}{, }
\affixside@note This macro puts any moveable sidenote text into the left or right sidenote box, depending on which margin it is meant to go in. It’s a very much stripped down version of \affixlin@num.
Before do it, we concatenate all moveable sidenotes of the line, using \sidenotesep as separator. It’s the result that we put on the sidenote.
\newcommand{\affixside@note}{% \
  \def\sidenotecontent@{}% \
  \numdef{\itemcount@}{0}% \
  \def\do##1{% \
    \ifnumequal{\itemcount@}{0}% \
      {\appto\sidenotecontent@{##1}}% Not print not separator before the 1st note \
      {\appto\sidenotecontent@{\sidenotesep ##1}}% \
    }% \
    \numdef{\itemcount@}{\itemcount@+1}% \
  %} \
\set@dlp@rbox × \set@dlpr@box{(lednums)}{(tag)}{(text)} puts (text) into the \l@dlp@rbox box.
\set@drp@box And similarly for the right side box. It is these boxes that finally get displayed in the margins.
We can put footnotes into minipages. The preparatory code has been set up earlier, all that remains is to ensure that it is available inside a minipage box. This requires some alteration to the kernel code, specifically the \@iiiminipage and \endminipage macros. We'll arrange this so that additional series can be easily added.

\@feetbeginmini These will be the hooks in \@iiiminipage and \endminipage They can be extended to handle other things if necessary.

\@feetendmini
\l@dedbeginmini These handle the initiation and closure of critical footnotes in a minipage environment.

\l@dedendmini

\l@dfambeginmini These handle the initiation and closure of familiar footnotes in a minipage environment.

\l@dfamendmini
\@iiiminipage  This is our extended form of the kernel \@iiiminipage defined in \texttt{ltboxes.dtx}.

\@iiiminipage  \begin{itemize}
\item This is our extended form of the kernel \@iiiminipage defined in \texttt{ltboxes.dtx}.
\item The next line is our addition to the original.
\item This is our extended form of the kernel \@iiiminipage defined in \texttt{ltboxes.dtx}.
\item The next line is our addition to the original.
\end{itemize}
\newcommand*{\l@dunboxmpfoot}{\vskip\skip\@mpfootins\normalcolor\footnoterule\ifparledgroup\ifl@dpairing\ifledRcol\dimgdef{\parledgroup@beforenotesR}{\parledgroup@beforenotesR+\skip\@mpfootins}\else\dimgdef{\parledgroup@beforenotesL}{\parledgroup@beforenotesL+\skip\@mpfootins}\fi\fi\fi\unvbox\@mpfootins}

\ledgroup This environment puts footnotes at the end, even if that happens to be in the middle of a page, or crossing a page boundary. It is a sort of unboxed, fixed width minipage.

\begin{ledgroupsized}{⟨pos⟩}{⟨width⟩}
This environment puts footnotes at the end, even if that happens to be in the middle of a page, or crossing a page boundary. It is a sort of unboxed, variable ⟨width⟩ minipage. The optional ⟨pos⟩ controls the sideways position of numbered text.

\end{ledgroupsized}
40 Indexing

Here's some code for indexing using page & line numbers.

First, ensure that imakeidx or indextools is loaded before eledmac.

\AtBeginDocument{%
  \unless\if@imakeidx%
    \@ifpackageloaded{imakeidx}{\led@error@ImakeidxAfterEledmac}{}%
  \fi%
  \unless\if@indextools%
    \@ifpackageloaded{indextools}{\led@error@indextoolsAfterEledmac}{}%
  \fi%
%
% \pagelinesep
% \edindexlab \c@labidx

In order to get a correct line number we have to use the label/ref mechanism.

These macros are for that.
\doedindexlabel  This macro sets an \edlabel.
\begin{verbatim}
newcommand{\doedindexlabel}{\stepcounter{labidx} edlabel{\edindexlab\thelabidx}}
\end{verbatim}

\thepageline  This macro makes up the page/line number combo from the label/ref.
\begin{verbatim}
newcommand{\thepageline}{\thepage\pagelinesep\xlineref{\edindexlab\thelabidx}}
\end{verbatim}

\thestartpageline, \theendpageline  These macros make up the page/line start/end number when the \edindex command is called in critical notes.
\begin{verbatim}
newcommand{\thestartpageline}{\l@dparsedstartpage\pagelinesep\l@dparsedstartline}
newcommand{\theendpageline}{\l@dparsedendpage\pagelinesep\l@dparsedendline}
\end{verbatim}

\if@edindex@fornote@true  This boolean test is switching at the beginning of each critical note, to allow indexing in this note.
\begin{verbatim}
newif\if@edindex@fornote@
prepare@edindex@fornote
\newcommand{\prepare@edindex@fornote}[1]{
\l@dp@rsefootspec#1|\
\@edindex@fornote@true\}
\end{verbatim}
\get@edindex@ledinnote@command  The \get@edindex@ledinnote@command macro defines a \@ledinnote@command command which is added as an attribute (text inserted after |) of the next index entry.

Consequently, we write the definition of the location reference attribute in the .xdy file.
\begin{verbatim}
newcommand{\get@edindex@ledinnote@command}{\iffindxyp% \edef\@ledinnote@command{ledinnote\thelabidx}% \iffindxyhyperref% \immediate\write\eledmac@xindy@out{(define-attributes ("ledinnote\thelabidx"))}\immediate\write\eledmac@xindy@out{(\text{mark-up-locref})}\immediate\write\eledmac@xindy@out{(\text{markup-locref})\immediate\write\eledmac@xindy@out{(\text{depth})\immediate\write\eledmac@xindy@out{}}% \end{verbatim}
If we do not use `xindy` option, `\@ledinnote@command` will produce something like `ledinnote{formattingcommand}`.

These macros are used to specify that an index reference points to a note. Arguments of `\ledinnote` are: #1 (optional): the label for the hyperlink, #2: command applied to the number, #3: the number itself.

\ledinnote\ledinnotehyperpage\ledinnotemark

These macros are used to analyse if a text to be indexed has a command after a `|`.

\get@index@command

This macro is used to analyse if a text to be indexed has a command after a `|`.
The \texttt{memoir} class provides more flexible indexing than the standard classes. We need different code if the \texttt{memoir} class is being used, except if \texttt{imakeidx} or \texttt{indextools} is used.

\section*{40.1 Memoir compatibility}

\texttt{create@edindex@for@memoir} define the \texttt{edindex} command and related tool when:

1. Memoir class is used.
2. AND \texttt{imakeidx} is not used.
3. AND \texttt{indextools} is not used.

Need to add the definition of \texttt{edindex} to \texttt{makeindex}, and initialise \texttt{edindex} to do nothing. In this case \texttt{edindex} has an optional argument. We use the hook provided in \texttt{memoir} v1.61.

\texttt{\textbackslash l@d@index[\texttt{file}]} is the first stage of \texttt{edindex}, handling the \texttt{idx} file. This a virtually a verbatim copy of \texttt{memoir}'s \texttt{\@index}, the change being calling \texttt{\@wrindex@\texttt{memoir}} instead of \texttt{\@wrindex@\texttt{\textbackslash indexgroup}}.
\doedindexlabel
\begingroup
\@sanitize
\l@d@wrindexm@m}
\l@d@wrindexm@m[item] writes the idx file name and the indexed item to the aux file. These are almost verbatim copies of memoir's \@wrindexm@m and \@wrindexhyp.
\newcommand{\l@d@wrindexm@m}[1]{\l@d@@wrindexhyp##1||\}
\def\l@d@@wrindexhyp##1|##2|##3\{}%\ifshowindexmark\@showidx{##1}\fi
\ifx\##2\%
\if@edindex@fornote@%\protected@write\@auxout{}%{\string\@@wrindexm@m{\@idxfile}{##1|(ledinnotehyperpage}{\thestartpageline}}%\protected@write\@auxout{}%{\string\@@wrindexm@m{\@idxfile}{##1|)ledinnotehyperpage}{\theendpageline}}%\else%
\protected@write\@auxout{}%
\else
\protected@write\@auxout{}%
\if@edindex@fornote@%\protected@write\@auxout{}%
\if\Hy@temp@A\HyInd@ParenLeft
\if\Hy@temp@A\HyInd@ParenLeft\protected@write\@auxout{}%
\{\string\@wrindexm@m{\@idxfile}{##1|ledinnotehyperpage}{\thestartpageline}}%\protected@write\@auxout{}%
\{\string\@wrindexm@m{\@idxfile}{##1|ledinnotehyperpage}{\thestartpageline}}%
\else%
\protected@write\@auxout{}%
\if\Hy@temp@A\HyInd@ParenLeft
\protected@write\@auxout{}%
\{\string\@wrindexm@m{\@idxfile}{##1|##2hyperpage}{\thepageline}}%
\else%
\protected@write\@auxout{}%
\{\string\@wrindexm@m{\@idxfile}{##1|##2}{\thepageline}}%
\fi
\fi
\else
\protected@write\@auxout{}%
\{\string\@wrindexm@m{\@idxfile}{##1|ledinnote{##2}}{\thepageline}}%
\protected@write\@auxout{}%
\{\string\@wrindexm@m{\@idxfile}{##1|ledinnote{##2}}{\thepageline}}%
\else%
\protected@write\@auxout{}%
\{\string\@wrindexm@m{\@idxfile}{##1|##2}{\thepageline}}%
\fi
\fi
\fi
\endgroup
\@esphack}
This finishes the memoir-specific code.
}
40.2 Normal setting

\create@edindex@notfor@memoir define the \edindex command and related tool when:

1. Memoir class is NOT used.
2. OR imakeidx is used.
3. OR indextools is used.

\def\create@edindex@notfor@memoir{
\@wredindex

Write the index information to the idx file.

\@wredindex
\newcommandx{\@wredindex}[2][1=\expandonce\jobname, usedefault]{% #1 = the index name, #2 = the text
\global\let\old@Rlineflag\Rlineflag%
\global\let\old@edindex@notfor@memoir\edindex%
\edef\Rlineflag{}%
\if\imakeidx%
\if\edindex@fornote@%
\IfSubStr[1]{##2}{|}{\get@index@command##2+}{\get@index@command##2|+}%
\get@edindex@ledinnote@command%
\expandafter\imki@wrindexentry{##1}{\@index@txt|\@ledinnote@command}{\thestartpageline}%
\expandafter\imki@wrindexentry{##1}{\@index@txt|\@ledinnote@command}{\theendpageline}%
\else%
\get@edindex@hyperref{##2}%
\imki@wrindexentry{##1}{\@index@txt\@edindex@hyperref}{\thepageline}%
\fi%
\else%
\if\edindex@fornote@%
\IfSubStr[1]{##2}{|}{\get@index@command##2+}{\get@index@command##2|+}%
\get@edindex@ledinnote@command%
\expandafter\protected@write\indexdir{}{\string\indexentry{\@index@txt|\@ledinnote@command}{\thestartpageline}{}%}
\expandafter\protected@write\indexdir{}{\string\indexentry{\@index@txt|\@ledinnote@command}{\theendpageline}{}%}
\else%
\protected@write\indexdir{}{\string\indexentry{##2}{\thepageline}{}%}
\fi%
\fi%
\endgroup
\global\let\Rlineflag\old@Rlineflag%
\@esphack}

Need to add the definition of \edindex to \makeindex, and initialise \edindex to do nothing.

\pretocmd{\makeindex}{%
\def\edindex{\@bsphack}
40.3 Choose the right variant

Then call \texttt{\create@edindex@for@memoir} or \texttt{\create@edindex@notfor@memoir} depending on the use of \texttt{memoir} and \texttt{imakeidx}

\begin{verbatim}
@ifclassloaded{memoir}{%
  @ifpackageloaded{imakeidx}{
    \create@edindex@notfor@memoir
  }{
    @ifpackageloaded{indextools}{
      \create@edindex@notfor@memoir
    }{
      \create@edindex@for@memoir
    }
  }
}{}
\end{verbatim}

40.4 hyperref compatibility

\texttt{\hyperlinkformat} \texttt{\hyperlinkformat} command is to be used to have both a internal hyperlink and a format, when indexing.

\begin{verbatim}
\newcommand{\hyperlinkformat}[3]{%
  \ifstrempty{#1}{\hyperlink{#2}{#3}}%
  \csuse{#1}{\hyperlink{#2}{#3}}%
}%%%%%%%%%%%%%%%%%
\end{verbatim}

\texttt{\hyperlinkR} \texttt{\hyperlinkR} command is to be used to create a internal hyperlink and \texttt{\Rlineflag}, when indexing.

\begin{verbatim}
\newcommand{\hyperlinkR}[2]{%
  \hyperlink{#1}{#2\Rlineflag}%
}%
\end{verbatim}

\texttt{\hyperlinkformatR} \texttt{\hyperlinkformatR} command is to be used to create a internal hyperlink, a format and a \texttt{\Rlineflag}, when indexing.

\begin{verbatim}
\newcommand{\hyperlinkformatR}[3]{%
  \hyperlinkformat{#1}{#2}{#3\Rlineflag}%
}%
\end{verbatim}
\get@edindex@hyperref is to be used to define the \@edindex@hyperref macro, which, in index, links to the point where the index was called with hyperref.

We have to disable temporary spaces to work through a xstring bug (or feature?)

\edef\temp@{%
\catcode`\ =9 %space need for catcode
\#1%
\catcode`\ =10 % space need for catcode
}

Now, we define \@edindex@hyperref if the hyperindex of hyperref is enabled.

...
This \thelabidx will be called back at the closing parenthesis, to have the same number in \@edindex@hyperref command that we had at the opening parenthesis. \@edindex@hyperref start by a closing parenthesis, then followed by elemdmacXXX where XXX is the \thelabidx of the opening \edindex.

\IfStrEq{@index@parenthesis}{)}\% 
\xdef\@edindex@hyperref{|)eledmac\csuse{xindyparenthesis@@index@txt}}\%
\global\csundef{xindyparenthesis@@index@txt}\%
Write in the .xdy file the attributes of the location.
\begin{verbatim}
\immediate\write\eledmac@xindy@out{\% (define-attributes (eledmac\thelabidx))^^J
\space\space(markup-locref^^J\eledmacmarkuplocrefdepth^^J
:open "\string\hyperlink\ifledRcol R\fi%\{\edindexlab\thelabidx\}%
:attr "eledmac\thelabidx"
)}\%
\end{verbatim}
And now, in any other case.
\begin{verbatim}
\else\%
\gdef\@index@txt{#1}\%
\gdef\@edindex@hyperref{}\%
\fi\%
\end{verbatim}

41 Macro as environment

The following is borrowed, and renamed, from the amsmath package. See also the CTT thread 'eeq and amstex', 1995/08/31, started by Keith Reckdahl and ended definitively by David M. Jones.

Several of the [math] macros scan their body twice. This means we must collect all text in the body of an environment form before calling the macro.

\emptytoks This is actually defined in the amsgen package.
\newtoks\emptytoks
\emptytoks
The rest is from `amsmath`.

\@denvbody \ A token register to contain the body.

\addtol@denvbody \ \addtol@denvbody{arg} adds arg to the token register \@denvbody.

\l@dcollect@body \ The macro \l@dcollect@body starts the scan for the \end{...} command of the current environment. It takes a macro name as argument. This macro is supposed to take the whole body of the environment as its argument. For example, given \cenv#1 {...} as a macro that processes #1, then the environment form, \begin{env} \end{env} would call \l@dcollect@body\cenv.

\l@dpush@begins \ When adding a piece of the current environment’s contents to \@denvbody, we scan it to check for additional \begin tokens, and add a ‘b’ to the stack for any that we find.

\l@dcollect@@body \ \l@dcollect@@body takes two arguments: the first will consist of all text up to the next \end command, and the second will be the \end command’s argument. If there are any extra \begin commands in the body text, a marker is pushed onto a stack by the \l@dpush@begins function. Empty state for this stack means we have reached the \end that matches our original \begin. Otherwise we need to include the \end and its argument in the material we are adding to the environment body accumulator.
There was a question on CTT about how to use \collect@body for a macro taking an argument. The following is part of that thread.

From: Heiko Oberdiek <oberdiek@uni-freiburg.de>  
Newsgroups: comp.text.tex  
Subject: Re: Using \collect@body with commands that take >1 argument  
Date: Fri, 08 Aug 2003 09:03:20 +0200

eed132@psu.edu (Evan) wrote:  
> I'm trying to make a new Latex environment that acts like the\colorbox command that is part of the color package. I looked through  
> the FAQ and ran across this bit about using the \collect@body command  
> that is part of AMSLaTeX:  
> http://www.tex.ac.uk/cgi-bin/texfaq2html?label=cmdasenv  
>  
> It almost works. If I do something like the following:  
> \newcommand{\redbox}{\colorbox{red}{#1}}  
> \makeatletter  
> \newenvironment{coloredbox}{\collect@body \redbox}{\collect@body \redbox}{}  

You will get an error message: Command \redbox already defined. Thus you must rename either the command \redbox or the environment name.

> \begin{coloredbox}{blue}  
> Yadda yadda yadda... this is on a blue background...  
> \end{coloredbox}  
> and can't figure out how to make the \collect@body take this.

> \collect@body {\colorbox{red}}  
> \collect@body {\colorbox{red}}

The argument of \collect@body has to be one token exactly.
% ignore spaces at begin and end of environment
\newenvironment{coloredboxII}{% 
def\next@{\mycoloredbox{#1}}% 
collect@body\next@
\}{% 
\newcommand\mycoloredbox[2]{% 
\colorbox{#1}{\ignorespaces#2\unskip}\}% 
}\% support of optional color model argument
\newcommand\coloredboxIII\endcsname{} 
def\coloredboxIII#1#{% 
@coloredboxIII{#1}% 
}% \newcommand\@coloredboxIII#1#2{% 
def\next@{\mycoloredboxIII{#1}{#2}}% 
collect@body\next@
}% \newcommand\mycoloredboxIII[3]{% 
\colorbox#1{#2}{\ignorespaces#3\unskip}\}% 
\makeatother
\begin{document}
Black text before
\begin{coloredbox}{blue}
Hello World
\end{coloredbox}
Black text after
Black text before
\begin{coloredboxII}{blue}
Hello World
\end{coloredboxII}
Black text after
Black text before
\begin{coloredboxIII}[rgb]{0,0,1}
Hello World
\end{coloredboxIII}
Black text after
\end{document}

Yours sincerely
Heiko <oberdiek@uni-freiburg.de>
42 Verse

This is principally Wayne Sullivan’s code and commentary from EDSTANZA \cite{Sul92}.

The macro \texttt{\textbackslash hangingsymbol} is used to insert a symbol on each hanging of verses. For example, in French typography the symbol is ‘\texttt{[}’. We obtain it by the next code:

\begin{verbatim}
\renewcommand{\hangingsymbol}{[\,}
\end{verbatim}

The \texttt{\textbackslash ifinstanza} boolean is used to be sure that we are in a stanza part.

\begin{verbatim}
\ifinstanza\newcommand*{\hangingsymbol}{[\,}\i
\end{verbatim}

The boolean \texttt{\textbackslash ifinserthangingsymbol} is set to TRUE when \textbackslash @lock is greater than 1, i.e. when we are not in the first line of a verse. The switch of \texttt{\textbackslash ifinserthangingsymbol} is made in \texttt{\do@line} before the printing of line but after the line number calculation.

\begin{verbatim}
\ifinserthangingsymbol\ifinstanza\hangingsymbol\fi\fi
\end{verbatim}

\texttt{\textbackslash\&} macro is going to be usurped. We need an alias in case an \& needs to be typeset in a stanza. Define it rather than letting it in case some other package has already defined it.

\begin{verbatim}
\newcommand*{\ampersand}{\textbackslash \&}
\end{verbatim}

A count register is allocated for counting lines in a stanza; also allocated is a dimension register which is used to specify the base value for line indentation; all stanza indentations are multiples of this value. The default value of \texttt{\textbackslash stanzaindentbase} is 20pt.

\begin{verbatim}
\chardef\body=\catcode`\0
\catcode`\0=11
\chardef\next=\catcode`\&
\catcode`\&=\active
\end{verbatim}
\newcount\stanza@count
\newlength{\stanzaindentbase}
\setlength{\stanzaindentbase}{20pt}

\strip@szacnt The indentation of stanza lines are non-negative integer multiples of the unit called \stanzaindentbase. To make it easier for the user to specify these numbers, some list macros are defined. These take numerical values in a list separated by commas and assign the values to special control sequences using \mathchardef. Though this does limit the range from 0 to 32767, it should suffice for most applications, including penalties, which will be discussed below.

\setstanzavalues In the original \setstanzavalues{sza}{...} had to be called to set the indents, and similarly \setstanzavalues{szp}{...} to set the penalties. These two macros are a convenience to give the user one less thing to worry about (misspelling the first argument). Since version 0.13, the \stanzaindentsrepetition counter can be used when the indentation is repeated every n verses. The \managestanza@modulo is a command which modifies the counter stanza@modulo. The command adds 1 to stanza@modulo, but if stanza@modulo is equal to the stanzaindentsrepetition counter, the command restarts it.

\setstanzaindents \setstanzapenalties \managestanza@modulo

\newcommand*{\setstanzaindents}{1}\{\setstanzavalues{sza}{#1}\}
\newcommand*{\setstanzapenalties}{1}\{\setstanzavalues{szp}{#1}\}
\newcounter{stanzaindentsrepetition}
\newcount\stanza@modulo
\newcommand*{\managestanza@modulo}{0}\{
\advance\stanza@modulo\@ne
\ifnum\stanza@modulo=\value{stanzaindentsrepetition}\斯坦za@modulo\@ne\fi
\}

\stanzaindent\stanzaindent*
The macro \stanzaindent, when called at the beginning of a verse, changes the indentation normally defined for this verse by \setstanzaindent. The starred version skips the current verse for the repetition of stanza indent.
Now we arrive at the main works. \stanza@line sets the indentation for the line and starts a numbered paragraph—each line is treated as a paragraph. \stanza@hang sets the hanging indentation to be used if the stanza line requires more than one print line. If it is known that each stanza line will fit on one print line, it is advisable to set the hanging indentation to zero. \sza@penalty places the specified penalty following each stanza line. By default, this facility is turned off so that no penalty is included. However, the user may initiate these penalties to indicate good and bad places in the stanza for page breaking.

\stanza@line Now we have the components of the \stanza macro, which appears at the start of a group of lines. This macro initializes the count and checks to see if hanging indentation and penalties are to be included. Hanging indentation suspends the line count, so that the enumeration is by verse line rather than by print line. If the print line count is desired, invoke \let\startlock=\relax and do the same for \endlock. Here and above we have used \xdef to make the stored macros take up a bit less space, but it also makes them more obscure to the reader. Lines of the stanza are delimited by ampersands &. The last line of the stanza must end with \&. For convenience the macro \endstanzaextra is included. The user may use this to add vertical space or penalties between stanzas.

As a further convenience, the macro \startstanzahook is called at the begin-
ning of a stanza. This can be defined to do something useful.

\let\startstanzahook\relax
\let\endstanzaextra\relax
\def\@startstanza[#1]{% 
\noexpand\instanzatrue\expandafter
\begingroup\startstanzahook%
\catcode`\&\active%
\global\stanza@count\@ne\stanza@modulo\@ne
\noexpand\ifnum\expandafter\noexpand
\csname sza@0\endcsname=\z@\let\noexpand\stanza@hang\relax
\let\noexpand\endlock\relax
\noexpand\else\interlinepenalty
\@M\rightskip\z@ plus 1fil\relax
\noexpand\fi
\noexpand\ifnum\expandafter\noexpand
\csname szp@0\endcsname=\z@
\let\sza@penalty\relax
\noexpand\fi% 
\def\falseverse{%
\noexpand\led@war@FalseverseDeprecated%
\global\advance\stanza@modulo-\@ne%
\global\advance\stanza@count-\@ne%
\relax&\leavevmode\skipnumbering}
\def&{\noexpand\@stopstanza}%
\def\stanza@line[#1]{%
\if\falseverse
\noexpand\newverse[]\noexpand[]%
\noexpand\endlock\noexpand\pend[#1]\noexpand\endgroup
\instanzafalse%
\endstanzaextra%
\else
\noexpand\if\noexpand\\noexpand\falseverse[]\noexpand[]%
\def\noexpand\&\noexpand\&\noexpand\endlock\noexpand\pend[#1]
\sza@penalty\global\advance\stanza@count\@ne\stanza@line[#2]%
\noexpand\fi%
\fi%
\unskip%
\endlock%
\pend[#1]%
\endgroup%
\instanzafalse%
\endstanzaextra%
}
\flagstanza
Use \flagstanza[len]{text} at the start of a line to put text a distance len before the start of the line. The default for len is \stanzaindentbase.
\newcommand*{\flagstanza}[2][\stanzaindentbase]{%
\unskip%
\endlock\pend[#1]\sza@penalty\global%
\advance\stanza@count\@ne\stanza@line[#2]%
}

The ampersand \& is used to mark the end of each stanza line, except the last, which is marked with \&. This means that \halign may not be used directly
within a stanza line. This does not affect macros involving alignments defined outside `\stanzatwo`. Since these macros usurp the control sequence `\&`, the replacement `\ampersand` is defined to be used if this symbol is needed in a stanza. Also we reset the modified category codes and initialize the penalty default.

```latex
\catcode`\&=\next
\catcode`@=\body
\% \let\ampersand=\&
\setstanzavalues{szp}{0}
```

43 Arrays and tables

This is based on the work by Herbert Breger in developing `tabmac.tex`.

The original `tabmac.tex` file was void of comments or any explanatory text other than the above notice. The algorithm is Breger’s. I have made some cosmetic changes to the original code and reimplemented some things so they are more LaTeX-like. All the commentary is mine, as are any mistakes or errors.

\@dtabnoexpands An extended and modified version of the original additional no expansions..
Macros to disable and restore familiar notes, to prevent them from printing multiple times in edtabularx and edarrayx environments.

\disable@familiarnotes
\restore@familiarnotes

\disable@sidenotes
\restore@sidenotes

The same, for side notes.
\disable@notes Disable/restore side and familiar notes.
\restore@notes
\newcommand{\disable@notes}{%
\disable@sidenotes%
\disable@familiarnotes%
}\%
\newcommand{\restore@notes}{%
\restore@sidenotes%
\restore@familiarnotes%
}\%
\l@dampcount \l@dampcount is a counter for the \& column dividers and \l@dcolcount is a counter for the columns. These were \texttt{\Undcount} and \texttt{\stellencount} respectively.
\newcount\l@dampcount
\l@dampcount=1\relax
\newcount\l@dcolcount
\l@dcolcount=0\relax
\hilfsbox Some (temporary) helper items.
\newbox\hilfsbox
\newskip\hilfsskip
\newbox\Hilfsbox
\newcount\hilfscount
\newdimen\dcoli
\newdimen\dcolii
\newdimen\dcoliii
\newdimen\dcoliv
\newdimen\dcolv
\newdimen\dcolvi
\newdimen\dcolvii
\newdimen\dcolviii
\newdimen\dcolix
\newdimen\dcolx
\newdimen\dcolxi
\newdimen\dcolxii
\newdimen\dcolxiii
\newdimen\dcolxiv
\newdimen\dcolxv
\newdimen\dcolxvi
\newdimen\dcolxvii
\newdimen\dcolxviii
\newdimen\dcolxix
\newdimen\dcolxx
\newdimen\dcolxxi

30 columns should be adequate (compared to the original 60). These are the column widths. (Originally these were German spelled numbers e.g., \texttt{\eins}, \texttt{\zwei}, etc).
\newdimen\dcoli
\newdimen\dcolii
\newdimen\dcoliii
\newdimen\dcoliv
\newdimen\dcolv
\newdimen\dcolvi
\newdimen\dcolvii
\newdimen\dcolviii
\newdimen\dcolix
\newdimen\dcolx
\newdimen\dcolxi
\newdimen\dcolxii
\newdimen\dcolxiii
\newdimen\dcolxiv
\newdimen\dcolxv
\newdimen\dcolxvi
\newdimen\dcolxvii
\newdimen\dcolxviii
\newdimen\dcolxix
\newdimen\dcolxx
\newdimen\dcolxxi
\newdimen\dcolxxii
\newdimen\dcolxxiii
\newdimen\dcolxxiv
\newdimen\dcolxxv
\newdimen\dcolxxvi
\newdimen\dcolxxvii
\newdimen\dcolxxviii
\newdimen\dcolxxix
\newdimen\dcolxxx
\newdimen\dcolerr % added for error handling
\l@dcolwidth
This is a cunning way of storing the columnwidths indexed by the column number
\l@dcolcount, like an array. (was \Dimenzuordnung)
\l@dsetmaxcolwidth
Sets the column width to the maximum value seen so far. (was \Dimenzuordnung)
\stepl@dcolcount
This increments the column counter, and issues an error message if it is too large.
\l@dsetmaxcolwidth
Sets the column width to the maximum value seen so far. (was \Dimenzuordnung)
\EDTEXT
We need to be able to modify the \edtext and \critext macros and also restore
\xedtext their original definitions.
\CRITEXT
\xcritext
\EDLABEL
We need to be able to modify and restore the \edlabel macro.
Macros supporting modification and restoration of `\index`.

`\index` replaces its delineated argument by `\relax` (was `\verschwinden`).

`\indexoptarg` replaces these two arguments (first is optional) by `\relax`.

```
\def\l@dgobbledarg #1/{\relax}
\newcommand*{\l@dgobbleoptarg}[2]{\relax%}
```

`\measuremcell` Measure (recursively) the width required for a math cell. (was `\messen`)

```
\def\measuremcell #1&{\ifx #1\iffalse% 5039\let\NEXT=\relax5040\else\l@dcheckcols% 5041\l@dcolcount=0% 5042\let\NEXT=\measuremcell1% 5043\fi% 5044\l@setmaxcolwidth% 5045\let\NEXT=\measuremcell1% 5046\fi\NEXT}
```
\measuretcell Measure (recursively) the width required for a text cell. (was \messentext)
\def\measuretcell #1&{\% 
  \ifx #1\%\% 
    \ifnum\l@dcolcount=0\relax \let\NEXT\relax \fi
  \else\l@dcheckcols\l@dcolcount=0\relax \let\NEXT\measuretcell\fi
  \else\setbox\hilfsbox = \hbox{#1}\% 
    \stepl@dcolcount \l@dsetmaxcolwidth \let\NEXT\measuretcell \fi\NEXT}

\measuremrow Measure (recursively) the width required for a math row. (was \Messen)
\def\measuremrow #1\%\%{\% 
  \ifx #1&\let\NEXT\relax \else \measuremcell #1&\%\%\%\% \let\NEXT\measuremrow \fi\NEXT}

\measuretrow Measure (recursively) the width required for a text row. (was \Messentext)
\def\measuretrow #1\%\%{\% 
  \ifx #1&\let\NEXT\relax \else \measuretcell #1&\%\%\%\% \let\NEXT\measuretrow \fi\NEXT}

\edtabcolsep The length \edtabcolsep controls the distance between columns. (was \abstand)
\newskip\edtabcolsep \global\edtabcolsep=10pt
\let\Next=\next
\newcommand{\variab}{\relax}

\l@dcheckcols Check that the number of columns is consistent. (was \tabfehlermeldung)
\newcommand*{\l@dcheckcols}{\% 
  \ifnum\l@dcolcount=1\relax \else \ifnum\l@dampcount=1\relax \else \ifnum\l@dcolcount=\l@dampcount\relax 
  \fi\fi\fi}
The original definition of \verteilen and friends (‘verteilen’ is approximately ‘distribute’) was along the lines:

\def\verteilen #1\&\{
  \ifx #1!
    \ifnum\l@dcolcount=0%
      \removelastskip
    \else
      \l@dcolcount=0%
    \fi
  \fi
\fi
\l@dampcount=\l@dcolcount
\fi}
\newcommand\l@dmodforcritext{\let\critext\relax
  \def\do##1{\global\csletcs{##1footnote}{l@dgobbledarg}}
  \dolistloop{\@series}
  \let\edindex\nulledindex
  \let\linenum\@gobble}
\newcommand\l@drestoreforcritext{\def\do##1{\csdef{##1footnote}{##1##2/{\csuse{##1@@footnote}{##1}{##2}}}
  \dolistloop{\@series}
  \let\edindex\xindex}
\newcommand\l@dnullfills{\def\edlabel##1{}%
  \def\edrowfill##1##2##3{}%
}
\newcommand\l@drestorefills{\def\edrowfill##1##2##3{\@EDROWFILL@{##1}{##2}{##3}}%}

\l@dmodforcritext Modify and restore various macros for when \critext is used.
\l@drestoreforcritext
\l@dmodforedtext Modify and restore various macros for when \edtext is used.
\l@drestoreforedtext
\l@dnullfills Nullify and restore some column fillers, etc.
\l@drestorefills

The original definition of \verteilen and friends (‘verteilen’ is approximately ‘distribute’) was along the lines:

\def\verteilen #1\&\{
  \ifx #1!
    \ifnum\l@dcolcount=0%
      \removelastskip
    \else
      \l@dcolcount=0%
    \fi
  \fi
\fi
\l@dampcount=\l@dcolcount
\fi}
where the lines
\let\critext=xcritext\let\Dfootnote=D@@footnote
\let\Afootnote=A@@footnote\let\Bfootnote=B@@footnote
\let\Cfootnote=C@@footnote\let\linenum=\@line@@num
\hilfsskip=\Dimenzuordnung
\advance\hilfsskip by -\wd\hilfsbox
\def\label##1{\xlabel{##1}}
\hskip\hilfsskip$\displaystyle{#1}$\hskip\edtabcolsep
\let\Next=rverteilen
\fi\Next

...were common across the several *verteilen* macros, and also
\def\footnoteverschw{%
\let\critext=relax
\let\Afootnote=verschwinden
\let\Bfootnote=verschwinden
\let\Cfootnote=verschwinden
\let\Dfootnote=verschwinden
\let\linenum=\@gobble}

\letsforverteilen Gathers some lets and other code that is common to the *verteilen* macros.
\setmcellright Typeset (recursively) cells of display math right justified. (was \rverteilen)
\def\setmcellright #1&{ \def\edlabel##1{} \let\edindex\nulledindex
\ifx #1\let\Next\relax
\else \l@dcolcount=0
\let\Next=\setmcellright
\fi
\else%
\disable@dtabfeet
\stepl@dcolcount
\disable@notes
\setbox\hilfsbox=\hbox{$\displaystyle{#1}$}
\restore@notes
\letsforverteilen
\hskip\hilfsskip$\displaystyle{#1}$
\hskip\edtabcolsep
\let\Next=\setmcellright
\fi\Next}

\settcellright Typeset (recursively) cells of text right justified. (was \rverteilentext)
\def\settcellright #1&{ \def\edlabel##1{} \let\edindex\nulledindex
\ifx #1\let\Next\relax
\else \l@dcolcount=0
\let\Next=\settcellright
\fi
\else%
\disable@dtabfeet
\stepl@dcolcount
\disable@notes
\setbox\hilfsbox=\hbox{#1}
\restore@notes
\letsforverteilen
\hskip\hilfsskip#1
\hskip\edtabcolsep
\let\Next=\settcellright
\fi\Next}

\setmcellleft Typeset (recursively) cells of display math left justified. (was \lverteilen)
\def\setmcellleft #1&{ \def\edlabel##1{} \let\edindex\nulledindex
\ifx #1\let\Next\relax
\else \l@dcolcount=0 \let\Next=\setmcellleft
\fi
\else \disable@dtabfeet
\fi\Next}
\settcellleft  Typeset (recursively) cells of text left justified. (was \lverteiltext)
\def\settcellleft #1&{\def\edlabel##1{}%
\let\edindex\nulledindex
\ifx #1\text\ifnum\l@dcolcount=0 \let\Next\relax%
\else\l@dcolcount=0%
\let\Next=\settcellleft%
\fi%
\else
\disable@dtabfeet%
\step1@dcolcount%
\disable@notes%
\setbox\hilfsbox=\hbox{$\displaystyle{#1}$}%
\restore@notes%
\letsforverteilen%
$\displaystyle{#1}$\hskip\hilfsskip\hskip\edtabcolsep%
\let\Next=\settcellleft%
\fi\Next}
\settcellcenter  Typeset (recursively) cells of display math centered. (was \zverteilen)
\def\settcellcenter #1&{\def\edlabel##1{}%
\let\edindex\nulledindex
\ifx #1\text\ifnum\l@dcolcount=0 \let\Next\relax%
\else\l@dcolcount=0%
\let\Next=\settcellcenter%
\fi%
\else
\disable@dtabfeet%
\step1@dcolcount%
\disable@notes%
\setbox\hilfsbox=\hbox{$\displaystyle{#1}$}%
\restore@notes%
\letsforverteilen%
#1\hskip\hilfsskip\hskip\edtabcolsep%
\let\Next=\settcellcenter%
\fi\Next}
\settcellcenter  Typeset (recursively) cells of text centered. (new)
\def\settcellcenter #1&{\def\edlabel##1{}%
\let\edindex\nulledindex

def\setmrowright #1\\{
   \ifx #1&\let\NEXT\relax
   \else\centerline{\setmcellright #1&\&\&}
   \let\NEXT=\setmrowright
   \fi\NEXT
\}

\settrowright Typeset (recursively) rows of right justified text. (was \rsetzentext)
\def\settrowright #1\\{
   \ifx #1&\let\NEXT\relax
   \else\centerline{\settcellright #1&\&\&}
   \let\NEXT=\settrowright
   \fi\NEXT
\}

\setmrowleft Typeset (recursively) rows of left justified math. (was \lsetzen)
\def\setmrowleft #1\\{
   \ifx #1&\let\NEXT\relax
   \else\centerline{\setmcellleft #1&\&\&}
   \let\NEXT=\setmrowleft
   \fi\NEXT
\}

\settrowleft Typeset (recursively) rows of left justified text. (was \lsetzentext)
\def\settrowleft #1\\{
   \ifx #1&\let\NEXT\relax
   \else\centerline{\settcelleft #1&\&\&}
   \let\NEXT=\settrowleft
   \fi\NEXT
\}
\setmrowcenter  Typeset (recursively) rows of centered math. (was \setzen)

\settrowcenter  Typeset (recursively) rows of centered text. (new)

\nullsetzen  (was \nullsetzen)

\edatleft  \edatleft[(math)]{(symbol)}{(len)} (combination and generalisation of original \seklam and \seklaml). Left \langle symbol \rangle, 2 \langle len \rangle high with prepended \langle math \rangle vertically centered.

\edatright  \edatright[(math)]{(symbol)}{(len)} (combination and generalisation of original \seklam and \seklaml). Right \langle symbol \rangle, 2 \langle len \rangle high with appended \langle math \rangle vertically centered.
\edvertline \edvertline\{\langle len \rangle\} vertical line \langle len \rangle high. (was \texttt{\vsestrich})
\newcommand{\edvertline}{[1]{\vbox to 8pt{\vss\hbox{\vrule height #1}\vfil}}}

\edvertdots \edvertdots\{\langle len \rangle\} vertical dotted line \langle len \rangle high. (was \texttt{\vsepunkte})
\newcommand{\edvertdots}{[1]{\vbox to 1pt{\vss\vbox to #1{\cleaders\hbox{$\m@th\hbox{.}$}\vbox to 0.5em{ }\vfil}}}}

I don’t know if this is relevant here, and I haven’t tried it, but the following appeared on CTT.

From: mdw@nsict.org (Mark Wooding)
Newsgroups: comp.text.tex
Subject: Re: Dotted line
Date: 13 Aug 2003 13:51:14 GMT

Alexis Eisenhofer <alexis@eisenhofer.de> wrote:
> Can anyone provide me with the LaTeX command for a vertical dotted line?

How dotted? Here’s the basic rune.
\newbox{linedotbox}
\setbox{linedotbox}=\vbox{...}
\leaders{\copy{linedotbox}}
\vskip 2in

For just dots, this works:
\setbox{linedotbox}=\vbox{\hbox{\normalfont.}\kern 2pt}

For dashes, something like
\setbox{linedotbox}=\vbox{\leaders{\vrule}\vskip 2pt\vskip 2pt}

is what you want. (Adjust the ‘2pt’ values to taste. The first one is
the length of the dashes, the second is the length of the gaps.)

For dots in mid-paragraph, you need to say something like
\lower{10pt}\vbox{\leaders{\copy{linedotbox}}\vskip 2in}

which is scungy but works.

-- [mdw]

\edfilldimen A length. (was \texttt{\klamdimen})
\newdimen\edfilldimen
\edfilldimen=0pt

\c@addcolcount A counter to hold the number of a column. We use a roman number so that we
can grab the column dimension from \texttt{\dcol}.
\newcounter{addcolcount}
\renewcommand{\theaddcolcount}{\roman{addcolcount}}
\l@dtabaddcols\l@dtabaddcols\{\{startcol\}\{endcol\}\} adds the widths of the columns \{startcol\} through \{endcol\} to \edfilldimen. It is a LaTeX style reimplementation of the original \@add@.

\newcommand{\l@dtabaddcols} [2] {\%\l@dcheckstartend\{#1\}{#2}\%\if\l@dstartendok\setcounter{addcolcount}{#1}\%\@whilenum \value{addcolcount}<#2\relax \do\{\advance\edfilldimen by \the \csname dcol\theaddcolcount\endcsname\%\advance\edfilldimen by \edtabcolsep\%\stepcounter{addcolcount}\}\%\\advance\edfilldimen by \the \csname dcol\theaddcolcount\endcsname\%\fi\%\if\l@dstartendok\l@dcheckstartend\l@dcheckstartend\{\\{startcol\}\{endcol\}\}\check that the values of \{startcol\} and \{endcol\} are sensible. If they are then \l@dstartendok is set TRUE, otherwise it is set FALSE.

\newif\if\l@dstartendok\newcommand{\l@dcheckstartend} [2] {\%\l@dstartendoktrue\\ifnum #1<\@ne\l@dstartendokfalse\%\led@err@LowStartColumn\%\fi\%\ifnum #2>30\relax\l@dstartendokfalse\%\led@err@HighEndColumn\%\fi\%\ifnum #1>#2\relax\l@dstartendokfalse\%\led@err@ReverseColumns\%\fi\%\}%\%}

\edrowfill\edrowfill\{\{startcol\}\{endcol\}\} fill fills columns \{startcol\} to \{endcol\} inclusive with \{fill\} (e.g. \hrulefill, \upbracefill). This is a LaTeX style reimplementation and generalization of the original \waklam, \Waklam, \waklamec, \wastricht and \wapunktel macros.

\newcommand*{\edrowfill} [3] {\%\l@dtabaddcols\{#1\}{#2}\%\hl@xt@ \the\l@dcolwidth{\hl@xt@ \the\edfilldimen{#3}\hss}\%\let\@edrowfill@ = \edrowfill\%\def\@EDROWFILL@ #1#2#3{\@edrowfill@ {#1}{#2}{#3}}\%}

\edbeforetab\edbeforetab\{\{text\}\{math\}\} puts \{text\} at the left margin before \edaftertab

\edbeforetab
\edaftertab
The macro \edbeforetab\{\{text\}\{math\}\} puts \{text\} at the left margin before
array cell entry \textit{\textit{math}}. Conversely, the macro $\textit{\textit{edaftertab}}\{\textit{\textit{math}}\}\{\textit{\textit{text}}\}$ puts \textit{\textit{text}} at the right margin after array cell entry \textit{\textit{math}}. $\textit{\textit{edbeforetab}}$ should be in the first column and $\textit{\textit{edaftertab}}$ in the last column. The following macros support these.

\begin{verbatim}
\leftltab \leftltab\{\textit{\textit{text}}\} for $\textit{\textit{edbeforetab}}$ in \ltab. (was \textit{\textit{linksltab}})
\newcommand{\leftltab}[1]{% #1 \hb@xt@\z@{\vbox{\edtabindent\moveleft\Hilfsskip\hbox{ #1}}\hss}}
\leftrtab \leftrtab\{\textit{\textit{math}}\}\{\textit{\textit{text}}\} for $\textit{\textit{edbeforetab}}$ in \rtab. (was \textit{\textit{linksrtab}})
\newcommand{\leftrtab}[2]{% #2 \hb@xt@\z@{\vbox{\edtabindent\moveleft\Hilfsskip\hbox{ \#1}}\hss}}
\leftctab \leftctab\{\textit{\textit{text}}\}\{\textit{\textit{math}}\} for $\textit{\textit{edbeforetab}}$ in \ctab. (was \textit{\textit{linksztab}})
\newcommand{\leftctab}[2]{% \setbox\hilfsbox=\hbox{#2}\l@dampcount=\l@dcolcount% \hb@xt@\z@{\vbox{\edtabindent\l@dcolcount=\l@dampcount% \moveleft\Hilfsskip\hbox{ #1}}\hss}}
\rightctab \rightctab\{\textit{\textit{math}}\}\{\textit{\textit{text}}\} for $\textit{\textit{edaftertab}}$ in \ctab. (was \textit{\textit{rechtsztab}})
\newcommand{\rightctab}[2]{% \setbox\hilfsbox=\hbox{\def\edlabel##1{}\l@dampcount=\l@dcolcount% \hb@xt@\z@{\vbox{\edtabindent\l@dcolcount=\l@dampcount% \moveleft\Hilfsskip\hbox{ #2}}\hss}}
\rightltab \rightltab\{\textit{\textit{math}}\}\{\textit{\textit{text}}\} for $\textit{\textit{edaftertab}}$ in \ltab. (was \textit{\textit{rechtsltab}})
\newcommand{\rightltab}[2]{% \setbox\hilfsbox=\hbox{\def\edlabel##1{}\l@dampcount=\l@dcolcount% \hb@xt@\z@{\vbox{\edtabindent\l@dcolcount=\l@dampcount% #1}}\hss}}
\\end{verbatim}
\rightrtab \rightrtab{(math)}{(text)} for \edafterrtab in \rtab. (was \rechtsrtab)

\newcommand{\rightrtab}[2]{%
\setbox\hilfsbox=hbox{\def\edlabel##1{}%
\disablel@dtabfeet$\displaystyle{#1}$}%
\advance\Hilfsskip by-\wd\hilfsbox%
\setbox\hilfsbox= \hbox{\def\edlabel##1{}%}
\disablel@dtabfeet$\displaystyle{#2}$}%
\advance\Hilfsskip by-\wd\hilfsbox%
\advance\Hilfsskip by\edtabcolsep%
\moveright\Hilfsskip\hbox{ #2}\hss%
}

\rtab text
\rtabtext{(body)} typesets (body) as a tabular with the entries right justified. (was \rtabtext)

\newcommand{\rtab}{(body)} typesets (body) as an array with the entries right justified. (was \rtab)

\edbeforertab \rtab (Here and elsewhere, \edbeforertab and \edafterrtab were originally \davor and \danach) The original \rtab and friends included a fair bit of common code which I have extracted into macros.

The process is first to measure the (body) to get the column widths, and then in a second pass to typeset the body.
\newcommand{\rtabtext}[1]{% 
  \l@nullfills 
  \measuretbody{#1}% 
  \l@restorefills 
  \settrowright \l@\&\l@\%
  \enablel@tabfeet}

\measuretbody \( \text{measures the tabular } \langle \text{body} \rangle \).

\newcommand{\ltab}{% 
  \l@nullfills 
  \l@\def\edbeforetab##1##2{\leftltab{##1}{##2}}\l@\restore@notes \l@\setmrowleft \l@\&\l@\%
  \enablel@tabfeet \l@\measurembody{#1}\l@\restorefills}

Array with entries left justified. (was \ltab)

\newcommand{\ltabtext}{% 
  \l@nullfills 
  \measuretbody{#1}% 
  \l@restorefills 
  \settrowleft \l@\&\l@%
  \enablel@tabfeet}

Tabular with entries left justified. (was \ltabtext)

\newcommand{\ctab}{% 
  \l@nullfills 
  \l@\def\edbeforetab##1##2{\leftctab{##1}{##2}}\l@\setmrowleft \l@\&\l@%
  \enablel@tabfeet \l@\measuretbody{#1}\l@\restorefills}

Array with centered entries. (was \ztab)
\ctabtext Tabular with entries centered. (new)
\newcommand{\ctabtext}[1]{
  \l@dnullfills
  \measuretbody{#1}%
  \l@drestorefills
  \settrowcenter #1\&\%
  \enablel@dtabfeet}

\spreadtext (was \breitertext)
\newcommand{\spreadtext}[1]{%\l@dcolcount=\l@dampcount%
  \hb@xt@ \the\l@dcolwidth{\hbox{#1}\hss}}
\spreadmath (was \breiter, ‘breiter’ = ‘broadly’)
\newcommand{\spreadmath}[1]{%\l@dcolcount=\l@dampcount%
  \hb@xt@ \the\l@dcolwidth{\hbox{$\displaystyle{#1}$}\hss}}

I have left the remaining TABMAC alone, apart from changing some names. I’m not yet sure what they do or how they do it. Authors should not use any of these as they are likely to be mutable.

\tabellzwischen (was \tabellzwischen)
\def{\tabellzwischen #1&}{%\l@dcolcount=\l@dampcount%
  \ifx #1\relax \let\NEXT\relax
  \else \stepl@dcolcount%
  \l@dcolwidth = #1 mm
  \let\NEXT=\tabellzwischen
  \fi \NEXT }

\edatabell For example \edatabell 4 & 19 & 8 \ specifies 3 columns with widths of 4, 19, and 8mm. (was \atabell)
\def{\edatabell #1\&\&}{%}
\Setzen (was \Setzen, ‘setzen’ = ‘set’)
\def{\Setzen #1&}{%\l@dcolcount=\l@dampcount%
  \ifx #1\relax \let\NEXT=\relax
  \else \stepl@dcolcount%
  \l@dcolwidth \let\TABLEskip=\l@dcolwidth
  \EDTAB #1}
\let\NEXT=\Setzen
\fi\NEXT}

\EDATAB (was \ATAB)
\def\EDATAB #1\{
  \ifx #1\Relax
    \centerline{\Setzen #1\relax&}
    \let\Next\relax
  \else
    \centerline{\Setzen #1&\relax&}
    \let\Next=\EDATAB
  \fi\Next}

\edatab (was \atab)
\newcommand{\edatab}[1]{{
  \variab
  \EDATAB #1\\Relax\\}
}

\HILFSkip  More helpers.
\Hilfsskip\newskip\HILFSkip
\newskip\Hilfsskip

\EDTABINDENT (was \TABINDENT)
\def\EDTABINDENT{\{
  \ifnum\l@dcolcount=30
    \let\NEXT\relax\l@dcolcount=0
  \else
    \stepl@dcolcount%
    \advance\Hilfsskip by\l@dcolwidth%
    \ifdim\l@dcolwidth=0pt
      \advance\hilfscount\@ne
    \else
      \advance\Hilfsskip by \the\hilfscount\edtabcolsep%
    \fi%
    \let\NEXT=\EDTABINDENT%
  \fi
  \fi\Next%
}

\edtabindent (was \tabindent)
\def\edtabindent{\{
  \l@dcolcount=0\relax
  \Hilfsskip=0pt
  \hilfscount=1\relax
  \EDTABINDENT%
  \hilfsskip=\hsize%
  \advance\hilfsskip -\Hilfsskip%
  \Hilfsskip=0.5\hilfsskip%
  }

\EDTAB (was \TAB)
\def\EDTAB #1|#2|{\{
  \let\Next=\EDTABINDENT

\setbox\tabhilfbox=\hbox{\displaystyle{#1}#2}%
Further helpers.

The 'environment' forms for \ltab, ctab and rtab.

The 'environment' forms for \ltabtext, ctabtext and rtabtext.

Here's the code for enabling \edtext (instead of \critext).

Declarations for using \critext{}.../ or using \edtext{}{} inside tabulers.

The default at this point is for \edtext.
44 Section’s title commands

44.1 Deprecated commands

\initnumbering@sectcmd defines \ledxxx commands. These commands are deprecated. It also defines quotation environment. Note: this assumes that the user didn’t change \chapter. If he did, he should redefine \initnumbering@sectcmd.

\newcommand{\initnumbering@sectcmd}{
\newcommand{\ledsection}[2]{%\led@war@ledxxxDeprecated{section}\leavevmode\pend\vspace{3.5ex \plus 1ex \minus .2ex}\ifl@dpairing\else\skipnumbering\fi%%
\leavevmode\ifledsecnolinenumber\skipnumbering\fi\section[#1]{##2}\leavevmode\vspace{2.3ex \plus.2ex}\skipnumbering\pend%%
\vspace{-2\parskip}\vspace{-2\baselineskip}%%
\ifautopar\else\pstart\fi
}}

\newcommand{\ledsubsection}[2]{%\led@war@ledxxxDeprecated{subsection}\leavevmode\pend\vspace{3.5ex \plus 1ex \minus .2ex}\ifl@dpairing\else\skipnumbering\fi%%
\leavevmode\ifledsecnolinenumber\skipnumbering\fi\subsection[#1]{##2}\leavevmode\vspace{1.5ex \plus .2ex}\skipnumbering\pend%%
\vspace{-2\parskip}\vspace{-2\baselineskip}%%
\ifautopar\else\pstart\fi
}}

\newcommand{\ledsubsubsection}[2]{%\led@war@ledxxxDeprecated{subsubsection}\leavevmode\pend\vspace{3.5ex \plus 1ex \minus .2ex}\ifl@dpairing\else\skipnumbering\fi%%
\leavevmode\ifledsecnolinenumber\skipnumbering\fi\subsubsection[#1]{##2}\leavevmode\vspace{1.5ex \plus .2ex}\skipnumbering\pend%%
\vspace{-2\parskip}\vspace{-2\baselineskip}%%
\ifautopar\else\pstart\fi
}}

\newcommand{\ledchapter*}[1]{%\led@war@ledxxxDeprecated{chapter}\leavevmode\pend\vspace{3.5ex \plus 1ex \minus .2ex}\ifl@dpairing\else\skipnumbering\fi%%
\leavevmode\ifledsecnolinenumber\skipnumbering\fi\section*{##1}\leavevmode\vspace{2.3ex \plus.2ex}\skipnumbering\pend%%
\vspace{-2\parskip}\vspace{-2\baselineskip}%%
\ifautopar\else\pstart\fi
}}

\newcommand{\ledsubsection*}[1]{%\led@war@ledxxxDeprecated{subsection}\leavevmode\pend\vspace{3.5ex \plus 1ex \minus .2ex}\ifl@dpairing\else\skipnumbering\fi%%
\leavevmode\ifledsecnolinenumber\skipnumbering\fi\subsection*{##1}\leavevmode\vspace{1.5ex \plus .2ex}\skipnumbering\pend%%
\vspace{-2\parskip}\vspace{-2\baselineskip}%%
\ifautopar\else\pstart\fi
}}

\newcommand{\ledsubsubsection*}[1]{%\led@war@ledxxxDeprecated{subsubsection}\leavevmode\pend\vspace{3.5ex \plus 1ex \minus .2ex}\ifl@dpairing\else\skipnumbering\fi%%
\leavevmode\ifledsecnolinenumber\skipnumbering\fi\subsubsection*{##1}\leavevmode\vspace{1.5ex \plus .2ex}\skipnumbering\pend%%
\vspace{-2\parskip}\vspace{-2\baselineskip}%%
\ifautopar\else\pstart\fi
}}

\newcommand{\quotation}{\quote}

\newcommand{\endquotation}{\endquote}
44.1 Deprecated commands

\catcode`\@=11
\catcode`\[=11
\catcode`\]=11
\mathchardef\miti=600A
\def\miti@avec#1\@avec\miti@avec#1\miti@avec#1\@avec
\ifx\@\miti\@\miti\@\miti\else\miti@avec#1\fi

\newcommand\ledchapter[2][%}
The \ledsectnotoc only disables the \addcontentsline macro.

\ledsectnomark The \ledsectnomark only disables the \chaptermark, \sectionmark and \subsectionmark macros.

44.2 New commands : \eledxxx

The new system of \eledxxx commands to section text work like this:
44.2 New commands: \eledxxx

1. When one of these commands is called, \eledmac writes to an auxiliary files:
   - The section level.
   - The section title.
   - The side (when \eledpar is used).
   - The pstart where the command is called.
   - If we have starred version or not.

2. \eledmac adds the title of the section to pstart, as normal content. This is to enable critical notes.

3. When \LaTeX is run a other time, this file is read. That:
   - Adds the pstart number to a list of pstarts where a sectioning command is used.
   - Defines a command, the name of which contains the pstart number, and which calls the normal \LaTeX sectioning command.

4. This last command is called when the pstart is effectively printed.

   We do not define commands for \eledsection and related if the noeledsec option is loaded. We use etoolbox tests and not the ifxxx..else..fi structure to prevent problem of expansions with command after the ifxxx which contains fi.

\beforeeledchapter For technical reasons, not yet solved, page-breaking before chapters can’t be made automatically by eledmac. Users have to use \beforeeledchapter.

\if@eled@sectioning The boolean \if@eled@sectioning is set to true when a sectioning command is called by a \eledxxx command, and set to false after. It is used to enable/disable line number printing.

\leftmargin@eledsection and \rightmargin@eledsection are added by eledmac inside the code of sectioning command, in order to affix line numbers. They include tests for RTL languages.
We have to patch \LaTeX, book and memoir sectioning commands in order to:

- Disable \texttt{edtext} inside.
- Disable page breaking (for \texttt{chapter}).
- Add line numbers and sidenotes.

Unfortunately, Maïeul Rouquette was not able to try if memoir is loaded. That is why eledmac tries to define for both standard class and memoir class.

\begin{verbatim}
\chapter \@sect
\M@sect
\@makechapterhead
\@makechapterhead
\@makeschapterhead
\@sect
\@ssect
\catcode`#=12 % Space NEEDS by \catcode
\AtBeginDocument{%
\patchcmd{\chapter}{\clearforchapter}{%
\if@eled@sectioning\else%
\ifl@dprintingpages\else%
\clearforchapter%
\fi%
\fi%}
\pretocmd{\M@sect}...
\endverbatim
44.2 New commands: \eledxxx

\let\old@edtext=\edtext%
\let\edtext=\dummy@edtext@showlemma%
\}
\{}\}
\apptocmd{\M@sect}
{\let\edtext=\old@edtext}
{\}
\apptocmd{\M@sect}
\patchcmd{\M@sect}
{\hskip #3\relax}
{\hskip #3\relax%\print@rightmargin@eledsection%}
{\}
{\}
\patchcmd{\M@sect}
\patchcmd{\@mem@old@ssect}
{\hskip #1}
{\hskip #1%\print@leftmargin@eledsection%}
{\}
{\}
\patchcmd{\@mem@old@ssect}
{\hskip #3\relax}
{\hskip #3\relax%\print@leftmargin@eledsection%}
{\}
{\}
\patchcmd{\chapter}{\if@openright\cleardoublepage\else\clearpage\fi}{%\if@eled@sectioning\else%\ifl@dprintingpages\else%
%No clearpage inside a \eledsection: will keep critical notes from printing on the title page.
44 Section's title commands

\patchcmd{\@makechapterhead}{#1}{\print@leftmargin@eledsection
 #1\print@rightmargin@eledsection}{}
\patchcmd{\@makechapterhead}{\if@RTL\raggedleft\else\raggedright\fi}{\if@eled@sectioning\else\if@RTL\raggedleft\else\raggedright\fi\fi}{}
\patchcmd{\@makeschapterhead}{#1}{\print@leftmargin@eledsection
 #1\print@rightmargin@eledsection}{}
\pretocmd{\@sect}{\let\old@edtext=\edtext\let\edtext=\dummy@edtext@showlemma}{}
\apptocmd{\@ssect}{\let\old@edtext=\edtext\let\edtext=\dummy@edtext@showlemma}{}
44.2 New commands: \eledxxx

\appcmd{@ssect}
{\let\edtext=\old@edtext}
{}
{}

hyperref also redefines @@sect. That’s why, when manipulating arguments, we
patch @@sect and the same only if hyperref is not used. If it is, we patch the NR
commands.

\ifpackageloaded{nameref}{
\patchcmd{\NR@sect}
{#8}
{#8%
\print@rightmargin@eledsection%
}
{}
{}
\patchcmd{\NR@sect}
{\hskip #3\relax}
{\hskip #3\relax%
\print@leftmargin@eledsection%
}
{}
{}
\patchcmd{\NR@ssect}
{#5}
{#5%
\print@rightmargin@eledsection%
}
{}
{}
\patchcmd{\NR@ssect}
{\hskip #1}
{\hskip #1%
\print@leftmargin@eledsection%
}
{}
{}
}\%
Section's title commands

\patchcmd{\@sect}
{\hspace{#3}\relax}
{\hspace{#3}\relax\print@leftmargin@eledsection}
{}
{}
{}
{}
\patchcmd{\@ssect}
{#5}
{#5\print@rightmargin@eledsection}
{}
{}
{}
\patchcmd{\@ssect}
{\hspace{#1}}
{\hspace{#1}\print@leftmargin@eledsection}
{}
{}
%
\catcode`\#=6 \eled@sectioning@out
\eled@sectioning@out is the output file, to dump the pstarts where a sectioning command is used.
\newwrite\eled@sectioning@out
\noeledsec The \noeledsec command is deprecated, people should use the noeledsec package option.
\newcommand{\noeledsec}{% 
\led@war@noeledsecDeprecated% 
\global\noeled@sectrue% 
}%
\eledchapter And now, the user sectioning commands, which write to the file, and also add content as a "normal" line.
\eledsection
\eledsubsection*\newcommand{\eledchapter}{2}[]{% 
\eledsubsubsection*\led@war@noeled@sect\eledsection* 
\eledchapter*\ifledRcol%
\eledsection*\immediate\write\eled@sectioning@out{% 
\eledsubsection*\string\eled@chapter{#1}\unexpanded{#2}{\the\@dnumstartsR}%{R} 
\eledsubsubsection*
44.2 New commands: \eledxxx

\newcommand{\eledchapter}{#2}%
\ifledRcol%
  \immediate\write\eled@sectioningR@out{%\string\eled@section{#1}{\unexpanded{#2}}{\the\l@dnumpstartsR}{}{R}%
\else%
  \immediate\write\eled@sectioning@out{%\string\eled@chapter{#1}{\unexpanded{#2}}{\the\l@dnumpstartsL}{}{%
  \fi%
\}
\}
\newcommand{\eledsection}{#2}%
\ifledRcol%
  \immediate\write\eled@sectioningR@out{%\string\eled@section{#1}{\unexpanded{#2}}{\the\l@dnumpstartsR}{}{R}%
\else%
  \immediate\write\eled@sectioning@out{%\string\eled@section{#1}{\unexpanded{#2}}{\the\l@dnumpstartsL}{}{%
\fi%
\}
\newcommand{\eledsubsection}{#2}%
\ifledRcol%
  \immediate\write\eled@sectioningR@out{%\string\eled@subsection{#1}{\unexpanded{#2}}{\the\l@dnumpstartsR}{}{R}%
\else%
  \immediate\write\eled@sectioning@out{%\string\eled@subsection{#1}{\unexpanded{#2}}{\the\l@dnumpstartsL}{}{%
\fi%
\}
\newcommand{\eledsubsubsection}{#2}%
\ifledRcol%
  \immediate\write\eled@sectioningR@out{%\string\eled@subsubsection{#1}{\unexpanded{#2}}{\the\l@dnumpstartsR}{}{R}%
\else%
  \immediate\write\eled@sectioning@out{%\string\eled@subsubsection{#1}{\unexpanded{#2}}{\the\l@dnumpstartsL}{}{%
\fi%
\WithSuffix\newcommand{\eledchapter}{#2}%
\ifledRcol%
  \immediate\write\eled@sectioningR@out{%\string\eled@section{#1}{\unexpanded{#2}}{\the\l@dnumpstartsR}{}{R}%
\else%
  \immediate\write\eled@sectioning@out{%\string\eled@section{#1}{\unexpanded{#2}}{\the\l@dnumpstartsL}{}{%
\fi%
\}
The sectioning macros, called in the auxiliary file. They have five arguments:

\eledchapter
\eledsection
\eledsubsection
\eledsubsubsection
1. Optional arguments of \LaTeX sectioning command.
2. Mandatory arguments of \LaTeX sectioning command.
3. Pstart number.
4. Side: R if right, nothing if left.
5. Starred or not.

\begin{verbatim}
\def\eled@chapter#1#2#3#4#5{\%
  \ifstrempty{#4}{\%
    \ifstrempty{#1}{%
      \global\csdef{eled@sectioning@#3#5}{\let\edtext=\dummy@edtext@showlemma\chapter{#2}}%
      \global\csdef{eled@sectmark@#3#5}{\let\edtext=\dummy@edtext{}\chaptermark{#2}}%
    }{%
      \global\csdef{eled@sectioning@#3#5}{\let\edtext=\dummy@edtext@showlemma\chapter[#1]{#2}}%
      \global\csdef{eled@sectmark@#3#5}{\let\edtext=\dummy@edtext{}\chaptermark{#2}}%
    }%
  }{%
    \ifstrempty{#1}{%
      \global\csdef{eled@sectioning@#3#5}{\let\edtext=\dummy@edtext@showlemma\chapter*{#2}}%
    }{%
      \global\csdef{eled@sectioning@#3#5}{\let\edtext=\dummy@edtext*[#1]{#2}}%
    }%
  }%
  \listcsgadd{eled@sections#5@@}{#3}%
}\def\eled@section#1#2#3#4#5{\%
  \ifstrempty{#4}{\%
    \ifstrempty{#1}{%
      \global\csdef{eled@sectioning@#3#5}{\section{#2}}%
      \global\csdef{eled@sectmark@#3#5}{\let\edtext=\dummy@edtext{}\sectionmark{#2}}%
    }{%
      \global\csdef{eled@sectioning@#3#5}{\section[#1]{#2}}%
      \global\csdef{eled@sectmark@#3#5}{\let\edtext=\dummy@edtext{}\sectionmark{#1}}%
    }%
  }{%
    \ifstrempty{#1}{%
      \global\csdef{eled@sectioning@#3#5}{\section*{#2}}%
    }{%
      \global\csdef{eled@sectioning@#3#5}{\section*[#1]{#2}}%
    }%
  }%
  \listcsgadd{eled@sections#5@@}{#3}%
}\def\eled@subsection#1#2#3#4#5{\%
  \ifstrempty{#4}{\%
    \ifstrempty{#1}{%
      \global\csdef{eled@sectioning@#3#5}{\subsection{#2}}%
    }{%
      \global\csdef{eled@sectioning@#3#5}{\subsection[#1]{#2}}%
    }%
  }{%
    \ifstrempty{#1}{%
      \global\csdef{eled@sectioning@#3#5}{\subsection*{#2}}%
    }{%
      \global\csdef{eled@sectioning@#3#5}{\subsection*[#1]{#2}}%
    }%
  }%
  \listcsgadd{eled@sections#5@@}{#3}%
}\end{verbatim}

\end{verbatim}
Page breaking or no page breaking depending of specific lines

By default, page breaks are automatic. However, the user can define lines which will force page breaks, or prevent page breaks around one specific line. On the first run, the line-list file records the line number of where the page break is being changed (either forced, or prevented). On the next run, page breaks occur either before or after this line, depending on how the user sets the command. The default setting is after the line.

\normalpagebreak  \normalpagebreak  is an etoolbox list which contains the absolute line number of the last line, for each page.

\def\normalpagebreak{}

\prevpb  The \l@prevpb macro is a etoolbox list, which contains the lines in which page breaks occur (before or after). The \l@prevnopb macro is a etoolbox list, which
contains the lines with NO page break before or after.

\ledpb The \ledpb macro writes the call to \led@pb in line-list file. The \ledpbnum macro writes the call to \led@pbnum in line-list file. The \lednopb macro writes the call to \led@nopb in line-list file. The \lednopbnum macro writes the call to \led@nopbnum in line-list file.

\ledpb\newcommand{\ledpb}{\write\linenum@out{\string\led@pb}}
\newcommand{\ledpbnum}[1]{\write\linenum@out{\string\led@pbnum{#1}}}
\newcommand{\lednopb}{\write\linenum@out{\string\led@nopb}}
\newcommand{\lednopbnum}[1]{\write\linenum@out{\string\led@nopbnum{#1}}}

\ledpb The \ledpb adds the absolute line number in the \prev@pb list. The \ledpbnum adds the argument in the \prev@pb list. The \lednopb adds the absolute line number in the \prev@nopb list. The \lednopbnum adds the argument in the \prev@nopb list.

\ledpb\newcommand{\led@pb}{\listxadd{\l@prev@pb}{\the\absline@num}}
\newcommand{\led@pbnum}[1]{\listxadd{\l@prev@pb}{#1}}
\newcommand{\led@nopb}{\listxadd{\l@prev@nopb}{\the\absline@num}}
\newcommand{\led@nopbnum}[1]{\listxadd{\l@prev@nopb}{#1}}

\ledpbsetting The \ledpbsetting macro only changes the value of \led@pb@macro, for which the default value is before.

\ledpbsetting\def\led@pb@setting{before}
\newcommand{\ledpbsetting}[1]{\gdef\led@pb@setting{#1}}

\ledcheck@pb The \ledcheck@pb and \ledcheck@nopb are called before or after each line. They check if a page break must occur, depending on the current line and on the content of \@pb.

\ledcheck@pb\newcommand{\ledcheck@pb}{\xifinlist{\the\absline@num}{\l@prev@pb}{\pagebreak[4]}}
\newcommand{\ledcheck@nopb}{\%\xifStrEq{\led@pb@setting}{before}{\xifinlist{\the\absline@num}{\l@prev@nopb}{\%{\numdef{\abs@prevline}{\the\absline@num-1}}\xifinlist{\abs@prevline}{\normal@page@break}{\%{\\enlargethispage{\baselineskip}}}\%}}}
\xifStrEq{\led@pb@setting}{after}{\%}}}
46 Long verse: prevents being separated by a page break

The \texttt{\textbackslash heldnopbinverse} boolean is set to false by default. If set to true, eledmac will automatically prevent page breaks inside verse. The declaration is made at the beginning of the file, because it is used as a package option.

\texttt{\textbackslash check@pb@in@verse} The \texttt{\check@pb@in@verse} checks if a verse is broken in two page. If true, it adds:

- The absolute line number of the first line of the verse -1 in the \texttt{\textbackslash led@pb} list, if the page break must occur before the verse.
- The absolute line number of the first line of the verse -1 in the \texttt{\textbackslash led@nopb} list, if the page break must occur after the verse.

\texttt{\newcommand{\check@pb@in@verse}{\ifinstanza\iflednopbinverse\ifinserthangingsymbol% Using stanzas and enabling page breaks in verse control, while on a hanging verse.\fi\fi\fi\fi\fi\fi}}

47 The End

</code>
Appendix A  Some things to do when changing version

Appendix A.1  Migrating from edmac

If you have never used EDMAC, ignore this section. If you have used EDMAC and are starting on a completely new document, ignore this section. Only read this section if you are converting an original EDMAC document to use eledmac.

The package still provides the original \text command, but it is (a) deprecated, and (b) its name has been changed\(^\text{\textsuperscript{35}}\) to \texttt{critext}; use the \texttt{edtext} macro instead. However, if you do use \texttt{critext} (the new name for \texttt{text}), the following is a reminder.

Within numbered paragraphs, footnotes and endnotes are generated by forms of the \texttt{critext} macro:

\texttt{\texttt{critext}\{\texttt{lemma}\}\{\texttt{commands}\}/}

The \texttt{\texttt{lemma}} argument is the lemma in the main text: \texttt{critext} both prints this as part of the text, and makes it available to the \texttt{\texttt{commands}} you specify to generate notes. The / at the end terminates the command; it is part of the macro’s definition so that spaces after the macro will be treated as significant.

For example:

\texttt{I saw my friend \texttt{\texttt{critext}{Smith}}\texttt{\Afootnote{Jones C, D.}/ on Tuesday.}}

The lemma \texttt{Smith} is printed as part of this sentence in the text, and is also made available to the footnote that specifies a variant, \texttt{Jones C, D}. The footnote macro is supplied with the line number at which the lemma appears in the main text.

The \texttt{\texttt{lemma}} may contain further \texttt{critext} commands. Nesting makes it possible to print an explanatory note on a long passage together with notes on variants for individual words within the passage. For example:

\texttt{\texttt{\texttt{critext}{I saw my friend \texttt{\texttt{critext}{Smith}{\Afootnote{Jones C, D.}/ on Tuesday.}}}}\texttt{\Bfootnote{The date was July 16, 1954.}/ }}

However, \texttt{critext} cannot handle overlapping but unnested notes—for example, one note covering lines 10–15, and another covering 12–18; a \texttt{critext} that starts in the \texttt{\texttt{lemma}} argument of another \texttt{critext} must end there, too. (The \texttt{\texttt{lemma}} and \texttt{\texttt{linenum}} commands may be used to generate overlapping notes if necessary.)

\(^{35}\)A name like \texttt{\texttt{text}} is likely to be defined by other \LaTeX\ packages (it certainly is by the AMS packages) and it seems sensible to try and avoid clashes with other definitions.
The second argument of the \textmacro, \text{\textbackslash commands}, is the same as the second argument to the \textmacro.

It is possible to define aliases for \textmacro, which can be easier to type. You can make a single character substitute for \textmacro by saying this:
\textbackslash catcode\textbackslash \textless active
\textbackslash let\textless \textequal \textbackslash text

Then you might say \textbackslash \textless \text{Smith} \textbackslash variant \text{Jones}. This of course destroys the ability to use \textless in any new macro definitions, so long as it remains in effect; hence it should be used with care.

Changing the character at the end of the command requires more work:
\textbackslash catcode\textbackslash \textless active
\textbackslash def\textbackslash xtext\textless \textequal \textbackslash textmacro\textless \textequal \textbackslash text\textbackslash \text{\textbackslash \textless }\textbackslash \text{\textgreater }
\textbackslash let\textequal \textbackslash xtext

This allows you to say \textbackslash \textless \text{Smith} \textbackslash \text{\textbackslash \textless }\text{Afootnote} \text{\textgreater }\text{Jones}.\textgreater.

Aliases for \textmacro of the first kind shown here also can’t be nested—that is, you can’t use the alias in the text that forms the first argument to \textmacro. (See section 22 to find out why.) Aliases of the second kind may be nested without any problem.

If you really have to use \textmacro in any of the tabular or array environments, then \textmacrornot must not be used in the same environment. If you use \textmacro in one of these environments then you have to issue the declaration \textmacro beforehand. The declaration \textmacro must be issued to revert to the default assumption that \textmacrornot will be used.

Appendix A.2 Migration from ledmac to eledmac

In ledmac, some changes were made in the code to allow for easy customization. This can cause problems for people who have made their own customizations. The next sections explain how to correct this.

If you have created your own series using \textmacro and \textmacroX, you should use instead the \textnewseries command (see 5.7.1 p. 34). You must remove your \textmacro command.

If you have customized the \textmacroXfmt command, you should check if commands for display options (5.4 p. 25) and options in \textmacroX (5.1.2 p. 19) cannot do the same thing. If not, you can add a new ticket in Github to request a new function for doing this.36

If for some reason you do not want to make the modifications to use eledmac new functions, you can continue using your own \textmacroXfmt command, but you must replace:
\textbackslash renewcommand*\{\textmacroXfmt\}[3]

36\url{https://github.com/maieul/ledmac/issues}
Appendix A.3 Migration to eledmac 1.5.1

The version 1.5.1 corrects a bug with `stanzaindentsrepetition` (cf. §6.1 p. 33). This bug had two consequences:

1. `stanzaindentsrepetition` didn’t work when its value was greater than 2.
2. `stanzaindentsrepetition` worked wrong when its value was equal to 2.

So, if you used `stanzaindentsrepetition` with value equal to 2, you must change your `setstanzaindents`. Explanation:

\begin{verbatim}
\setcounter{stanzaindentsrepetition}{2}
\setstanzaindents{5,1,0}
\end{verbatim}

This code, in a version older than 1.5.1, made that the first verse had an indent of 0, the second verse of 1, the third verse of 0, the fourth verse of 1 etc.

But instead the code should have assigned the reverse: the first verse had an indent of 1, the second verse of 0, the third verse of 1, the fourth verse of 0 etc.

So version 1.5.1 corrected this bug. If you want to keep the older presentation, you must change:

\begin{verbatim}
\setcounter{stanzaindentsrepetition}{2}
\setstanzaindents{5,1,0}
\end{verbatim}

by:

\begin{verbatim}
\setcounter{stanzaindentsrepetition}{2}
\setstanzaindents{5,0,1}
\end{verbatim}

Appendix A.4 Migration to eledmac 1.12.0

The migration to eledmac 1.12.0 is easy:

- You must delete all the auxiliary files, and so one, make the normal three runs.
- If you have modified `\l@reg`, which is not advisable, you must rename it to `\@nl@reg`.
Anyway, there is another problem. If you have text in brackets just after \pstart or \pend, the text will be considered an optional argument of \pstart or \pend (see 4.2.2 p. 13). In this case, just add a \relax between \pstart/\pend and the brackets.

The version 1.12.0 adds a new better way to manage section titles inside numbered text. Please read § 14.2 (14.2 p. 49).

Appendix A.5 Migration to eledmac 17.1

The version change the default behavior of \pstartinfootnote. Henceforth, the pstart will be printed if footnote only for the section of text where you have called \numberpstarttrue.

We don’t see any reason to print it in other section. However, if you want to print the pstart number in all footnote, with or without \numberpstarttrue, you can use \pstartinfootnoteeverytime.

Appendix A.6 Migration to eledmac 1.21.0

Appendix A.6.1 \Xledsetnormalparstuff and \ledsetnormalparstuffX

The \ledsetnormalparstuff has been split in two different commands:

- \Xledsetnormalparstuff for critical notes;
- \ledsetnormalparstuffX for familiar notes.

The new commands take an optional argument which is the series letter. If you have redefined \ledsetnormalparstuff or commands which call them, you must make the appropriate change.

Appendix A.6.2 Endnotes

In any case, clean the .end file before the next run.

The previous version of eledmac had a bug: there were two spaces between the start page number and the start line number, but only one space between the end page number and the end line number.

Indeed, a spurious space was added after the first \printnpnum. This spurious space has been deleted. However, if you want to keep the previous spurious space, just load the package with the oldprintnpnumspace option.

If you have redefined \endprint, you must:

- Contact us to ask for the feature that required your hack, in order to avoid such a hack in the future.
- Use the new fifth argument.
- Add \xdef@currentseries{#4} at the beginning of your own command.
Appendix A.7  Migration to eledmac 1.22.0

The \ledinote commands takes now a first optional argument, which is the label for the hyperreference. If you have redefined it, change your redefinition, and check if you can avoid this redefinition by redefining only \ledinotemark.

Appendix A.8  Migration to eledmac 1.23.0

People must delete the numbered auxiliary file before new run after update of eledmac.
References


[Fai03] Robin Fairbairns. footmisc — a portmanteau package for customising footnotes in \LaTeX. February 2003. (Available from CTAN in macros/latex/contrib/footmisc)


Index

Numbers written in italic refer to the page where the corresponding entry is described; numbers underlined refer to the code line of the definition; numbers in roman refer to the code lines where the entry is used.

Symbols

\# ................................................................. 5725, 5913
\& ............................................................... 34, 4777, 4781, 4782, 4844, 4859, 4882, 4884
@@ledleftnote .................................................... 4928, 4936
@@ledrightnote .................................................... 4927, 4935
@@ledsidenote .................................................... 4929, 4937
@@line ........................................................... 2228
@@wrindexe0m .................................................... 4566, 4568, 4571, 4578, 4580, 4583, 4588, 4590, 4593
@EDROWFILL@ .................................................. 5120, 5335
Index

\@MM ................................................................. 2228, 4837, 4849
\@MM ................................................................. 1773
\@addswfalse ....................................................... 1119
\@addswtrue ......................................................... 1121, 1123, 1127
\@adv ................................................................. 655, 912
\@arabic ............................................................. 1199
\@auxout ............................................................ 3900, 3913, 3961, 3964, 3967, 3970, 4565, 4567, 4570, 4577, 4579, 4582, 4587, 4589, 4592
\@backslashchar ..................................................... 4721
\@botlist ............................................................ 3848, 3850
\@cclv ............................................................... 3718, 3722, 3723, 3846, 3847, 3875
\@chapter .......................................................... 5628
\@checkend ........................................................ 4759
\@colht ............................................................. 3700, 3851, 3863
\@colroom ........................................................ 3851
\@combinefloats .................................................... 3695
\@currentlabel ..................................................... 1242, 2480, 2619, 2692, 2814
\@currentseries ................................................... 263
\@currentenvir ..................................................... 4743, 4746, 4747
\@currlist ........................................................ 3852, 3855
\@dbldeferlist ..................................................... 3861, 3866, 3868
\@dblfloatplacement ............................................... 3865
\@dbltoplist ....................................................... 3861, 3862
\@deferlist ........................................................ 3848, 3857, 3858
\@doclearpage ...................................................... 3822
\@edindex@fornote@true ........................................... 4478
\@edindex@hyperref ............................................... 4612, 4665
\@edrowfill@ ........................................................ 5835
\@eled@sectioningtrue .............................................. 794, 818–820, 822–826, 828, 830, 969, 974, 978, 980, 982, 985, 987, 989, 1021, 1028, 1059, 1115, 3258, 3951
\@ehb ............................................................... 3854
\@ehd ............................................................... 209, 212, 215, 218
\@eled@sectioningfalse ............................................ 1363
\@eled@sectioningtrue .............................................. 1361
\@emptytoks ........................................................ 4734, 4744
\@first ............................................................ 3419, 3421
\@firstoftwo ........................................................ 1155, 1172
\@fnpos ........................................................... 2901, 3736, 3739
\@footnotemark ..................................................... 2403
\@footnotetext ................................................... 4317, 4322, 4324, 4328, 4414, 4441
\@f freelist ........................................................ 3693
\@gobble ........................................................... 27, 801, 959, 960, 3074, 4756, 4930–4932, 5099, 5100, 5683–5685
\@gobblefive ........................................................ 222, 3185, 3374
\@gobblefour ....................................................... 220
\@gobbles three ..................................................... 220, 807, 5681
\@h ................................................................. 2226
\@hilfs@count ....................................................... 5035
\@idxfile .......................................................... 4555, 4566, 4568, 4571, 4578, 4580, 4583, 4588, 4590, 4593
Index

\@ifclassloaded .......................... 67, 2402, 3786, 3811, 4642
\@ifnextchar .......................... 5020
\@ifpackageloaded ...................... 70, 72, 3715, 4457, 4460, 4643, 4646, 5845
\@iiiminiptext .......................... 4367
\@iiiparbox .......................... 4394
\@index@command 4490, 4500, 4508, 4514, 4516–4518, 4521–4523, 4676, 4680, 4719, 4721
\@index@command@false .................. 4517, 4518, 4522, 4523
\@index@parenthesis .................. 4515, 4519, 4524, 4675, 4679, 4700, 4706
\@index@txt .......................... 4513, 4608, 4609, 4612, 4619, 4622, 4702, 4708, 4709, 4729
\@indexfalse .......................... 4618, 4621, 4625
\@inputcheck .......................... 536
\@insert .......................... 1658–1660, 1694–1696
\@k ................................ 2226
\@kludgeins .......................... 3697, 3783
\@lab .................................. 805, 3891, 3904, 3946
\@latexerr .......................... 3854
\@led@extranofeet .......................... 3808, 3818, 3840
\@led@nofootfalse .......................... 3825, 3830, 3835
\@led@nofoottrue .......................... 3823
\@led@testifnofoot .......................... 3822
\@ledinnote@command 4482, 4507, 4608, 4609, 4619, 4622
\@lemma .................................. 806, 1005, 1007, 1058
\@lemmacommand@false .......................... 975, 1023
\@lemmacommand@true .......................... 1051
\@line@num .................................. 5031, 5129
\@listdepth .......................... 4380
\@lock .................................. 283, 516, 602, 604, 606, 619, 722, 723, 725, 726, 742, 743, 745, 1322, 1404, 1464, 1466, 1467, 1469, 1575, 1590, 1592, 1594
\@lopl .................................. 639, 801
\@lopR .................................. 639
\@makechapterhead .......................... 5624–5627, 5725
\@makecol .......................... 3790
\@makecolumn .......................... 3857, 3858, 3866, 3868
\@makeschapterhead .......................... 5620–5623, 5725
\@makespecialcolbox .......................... 3698
\@maxdepth .......................... 3713, 3721
\@minipagetrue .......................... 3812
\@minipagemaxfeetfalse .......................... 3814, 3815
\@midlist .......................... 5725
\@minipagefalse .......................... 3693, 3694
\@minipagerestore .......................... 4391
\@minus .................................. 4381
\@minusbullet .................................. 3248, 3320, 5558, 5566, 5574, 5582, 5590, 5598
\@midlist .......................... 5725
\@minusbullet .................................. 3248, 3320, 5558, 5566, 5574, 5582, 5590, 5598
Index

\@mpargs ................................................................. 4371, 4394
\@mpfn ............................................................... 4377, 4413, 4440
\@mpfnpos .......................................................... 2901, 4316, 4319
\@mpfootins ....................................................... 4387, 4397, 4403, 4405, 4409, 4419, 4446
\@mpfootnotetext ............................................. 4378, 4414, 4441
\@mplistdepth ................................................... 4380
\@nameuse .............................................................. 465, 467,
1779, 1780, 2014, 2131, 2132, 2177, 2291, 2365, 2447, 2451, 2453, 2462, 2465,
2466, 2472, 2481, 2485, 2489, 2515, 2520, 2538, 2550, 2556, 2616, 2620, 2629,
2637, 2689, 2693, 2702, 2710, 2779, 2792, 2800, 2801, 2806, 2815, 2819, 2832,
3002, 3003, 3005, 3006, 3011, 3798, 3799, 3801, 3802, 3804, 3806, 4343, 4345
\@new ............................................................. 3406–3409, 3413–3416
\@next@page ......................................................... 856, 857
\@nl ................................................................. 577, 857, 859, 861, 868, 872, 875
\@nl@reg ............................................................ 527
\@nobreakfalse ...................................................... 1214, 1360
\@nobreaktrue ...................................................... 1214, 1218, 1360
\@noeled@sectrue .................................................. 22, 5917
\@noneed@Footnotefalse ........................................... 1021
\@noneed@Footnotetru ............................................. 3359
\@nowindex ............................................................ 4554
\@oldnbreak .......................................................... 1214, 1216, 1271
\@opcol .......................................................... 3858, 3876
\@opxtrafeetii .................................................... 3748, 3749, 3797
\@outputbox ....................................................... 2970, 2971, 3000, 3001, 3700, 3702, 3703, 3718, 3720, 3746, 3747
\@outputpage ........................................................ 3867
\@parboxrestore .................................................... 1784, 2470, 4376
\@patchforledchapter ............................................. 5555
\@pboxswfalse ....................................................... 4369
\@pend ............................................................. 639
\@pendR ............................................................ 639
\@percentchar ........................................................ 3368
\@plus ............................................................... 1796, 1805, 3248,
3320, 5558, 5560, 5566, 5568, 5574, 5576, 5582, 5584, 5590, 5592, 5598, 5600
\@ref ............................................................ 788, 883, 892
\@ref@reg .......................................................... 790
\@reinserts .......................................................... 3791
\@schapter ........................................................... 5629
\@second ............................................................. 3420, 3421
\@secondoftwo ....................................................... 1156, 1173
\@sect ............................................................. 5725
\@series ............................................................. 530, 534, 2979, 2992, 3201, 3203,
3400, 3405, 3408, 3409, 3412, 3414, 3416, 3419, 3420, 3431, 3451, 3763, 3776,
3817, 3839, 4327, 4350, 4357, 4364, 4914, 4922, 5097, 5102, 5108, 5113, 5128
\@set ............................................................. 670, 917
\@setminipage ....................................................... 4382
\@showidx ............................................................ 4562
\@sect ............................................................. 5725
\@startstanza ...................................................... 4839
\@stopstanza ........................................................ 4839
Index

\\&sw ................................. 807, 1090, 1093, 1098
\&tag .................................. 968, 992, 1052, 2425, 2426, 3275, 3292, 3364
\&tempbbox ................................ 384, 3847, 4372, 4394
\&tempcnta ............................... 1115–1117, 1120, 1126, 1134, 1140, 1144
\&tempdima ................................ 3722, 4370, 4374
\&temploD ................................ 4281, 4283
\&temploN ................................ 4282, 4283
\&textbottom ................................ 3705
\&texttop .................................. 3701
\&tmp ..................................... 825, 827, 828, 1134, 1136, 1138, 1140, 1154–1156, 1171–1173
\&tmpp ................................... 826, 827
\&toplist .................................. 3848, 3849
\&whilenum ................................ 1116, 5310
\&whilesw ................................ 3858, 3867
\&wredindex ................................ 4601, 5027
\&x@sf .................................... 2396, 2399, 2407, 2413, 2437, 2443
\&x@loop .................................. 1692, 1699
\&x@ympar ................................ 4089
\\^ ........................................... 560
\u ........................................... 4667, 4669, 5343, 5348, 5356

A
\abs@line@verse .................................. 6130, 6131, 6134, 6135
\abs@prevline ................................ 6109, 6110
\absline@num .................................. 277, 515, 582, 585, 588, 636, 684, 687, 696, 710, 732, 757, 767, 784, 796, 854, 855, 864, 866, 1101, 1102, 1113, 1395, 1416, 1417, 1425, 1657, 6099, 6101, 6102, 6105, 6108, 6109, 6117, 6118, 6130, 6134
\absline@numR ................................ 754, 764, 781, 1108, 1109, 1136

Abu Kamil Shuja' b. Aslam .............................. 10
\actionlines@list ................................ 518, 542, 545, 552, 684, 687, 696, 710, 732, 757, 767, 784, 1447, 1450
\actionlines@listR ................................ 754, 764, 781
\actions@list .................................. 518, 546, 553, 685, 694, 698, 700, 712, 721, 734, 741, 758, 765, 785, 1451
\actions@listR .................................. 755, 765, 782
\add@inserts .................................. 1344, 1358, 1646
\add@inserts@next ................................ 1646
\add@penalties ................................ 1355, 1667
\addcontentsline ................................ 5681
\addfootins .................................. 3794
\addfootinsX ................................ 2996
\addtocounter ................................ 1272
\addtocontents .................................. 4738, 4760, 4762

Adelard II ........................................ 10
\advancelabel@refs .............................. 3898, 3911, 3919
\advanceline .................................. 17, 115, 118, 912, 940, 947, 960
\advantref .................................. 3688
\endnote ........................................ 20
\affixline@num .................................. 1332, 1490
\affixpstart@num .............................. 1340, 1605
Index

\affixside@note ........................................ 1344, 1358, 4268
\Afootnote ............................................. 19
\afterlemmaseparator .................................. 28
\afternote ............................................. 31
\afternumberinfootnote ................................ 27
\afterruleX ........................................... 31
\aftersymlinenum ...................................... 27
\afterXrule ........................................... 31
\allowbreak ............................................ 2284, 2354, 2630, 2703
\ampersand ............................................. 37, 4777, 4884
\applabel ............................................... 40, 145, 3950
\appref ............................................... 40, 4042
\apprefprefixmore ..................................... 40, 4042
\apprefprefixsingle .................................. 40, 4042
\apprefwithpage ....................................... 40, 4042
\appto .................................................... 4274, 4275
\apptocmd .............................................. 2403, 2422, 5622, 5626, 5628, 5629, 5744, 5828, 5840
\at@every@pend ........................................ 1274, 1280
\at@every@pstart ..................................... 1195, 1196, 1206
\AtBeginDocument .................................... 43, 2861, 3715, 3942, 3987, 4455, 5726
\AtEndDocument ........................................ 47
\AtEveryPend .......................................... 14, 1280
\AtEveryPstart ....................................... 14, 1194
\autopar ............................................... 14, 136, 329, 1208, 1276, 1280
\autopar@pausetrue .................................... 325
\autoparfalse .......................................... 315, 1290
\autopartrue .......................................... 1303

B
\ballast .................................................. 51
\ballast@count ........................................ 1411, 1414, 1419, 1667
Beeton, Barbara Ann Neuhaus Friend .................... 14
\beforeeledchapter .................................... 5688
\beforeeledchapter .................................... 5630
\beforelemmaseparator ................................ 28
\beforenotesX .......................................... 31
\beforenumberinfootnote ................................ 26
\bfootsymlinenum ...................................... 27
\bfootsymlinknotes .................................... 31
\beginnumbering ........................................ 12, 139, 245, 346, 1221, 1300
\beginnumberingR ...................................... 1295
\bendnote ............................................. 20
\bfootnote ............................................. 19
\bfseries .............................................. 1199
\bhooknoteX ........................................... 30
\bhookXendnote ......................................... 30
\bhookXnote ........................................... 30
\bbody .................................................. 1700, 1701, 4779, 4883
\bbodyfootmarkX ...................................... 42
\boolfalse ............................................. 820
Index

\booltrue ................................................. 1059
\box ................................................. 1387, 1389, 2126, 2141, 2208, 2227, 2796, 2810, 3718, 3847, 3875
\boxendlinenum@appref .................................... 4054
\boxfootnotenumbers ...................................... 3657, 3660
\boxlinenum ............................................. 27
\boxlinenumalign ......................................... 27
\boxmaxdepth ............................................ 3721
\boxstartlinenum@appref .................................. 4053
\boxsymlinenum ........................................... 27
\boxXendendlinenum@apprefwithpage ......................... 4057
\boxXendendlinenumalign .................................. 28
\boxXendlinenum ........................................... 28
\boxXendlinenumalign ..................................... 28
\boxXendstartlinenum@apprefwithpage ...................... 4056
\boxXendstartlinenumalign ................................ 28
Bredon, Simon ............................................... 10
Breger, Herbert ............................................ 7, 10, 222
Brey, Gerhard .............................................. 10
\brokenpenalty ............................................ 1285
Burt, John .................................................. 8
Busard, Hubert L. L. ....................................... 10
\bypage@false ............................................ 350, 366, 374
\bypage@true ............................................. 350, 358
\bypstart@false ........................................... 350, 359, 375
\bypstart@true ............................................ 350, 367
C
\c@addcolcount ............................................ 5304
\c@ballast .............................................. 1411, 1419
\c@firstlinenum .......................................... 415, 1511, 1513, 1516, 1518
\c@firstsublinenum ....................................... 419, 1498, 1500, 1503, 1505
\c@abidx .................................................. 4463
\c@linenumincrement .................................... 415, 1514, 1515
\c@mpfootnote ............................................ 4377, 4413, 4440
\c@page ................................................... 859, 861, 867, 870, 872, 875, 4147, 4155
\c@pstart .................................................. 1199, 3914, 3968, 3971
\c@pstartR ................................................ 3901, 3962, 3965
\c@sublinenumincrement ................................ 419, 1501, 1502
\Cendnote .................................................. 20
\centerline ............................................... 5240, 5245, 5251, 5256, 5262, 5267, 5489, 5491
\Cfootnote ................................................ 19
\ch@ck@l@ck ................................................. 1530, 1559
\ch@cks@ub2@l@ck ......................................... 1507, 1559
\ch@pt@c .................................................. 5607
\changes .................................................... 4078
\chapter ................................................... 5611, 5617, 5631, 5632, 5725, 6027, 6031, 6037, 6038
\chaptermark .............................................. 5683, 6028, 6032
\char ........................................................ 4777
\chardef .................................................. 5607, 6126
\check@pb@in@verse ....................................... 1327, 6126
3048, 3050, 3062, 3064–3066, 3137, 3139, 3145, 3147, 3150, 3152, 3160, 3241, 3242, 3274, 3291, 3298, 3337, 3339, 3345, 3428, 3429, 3447, 3448, 3467, 3480, 3612, 3618, 3627, 3629, 3631–3635, 3656, 3661, 3665, 3667, 3669, 3682, 3686, 3750, 3751, 3757–3759, 3773, 3774, 3814, 3815, 3829, 3834, 4033, 4034, 4340, 4348, 4363, 4534, 4537, 4540, 4655, 4708, 4910, 4911, 5101, 5704, 5720, 6065, 6069
\csxdef .................................................. 630, 1101, 1108, 1723, 1725, 1729, 1741, 1744, 1749, 1752, 1755, 2054, 2076, 2096, 2259, 2334, 2571, 2605, 2679, 2741, 3648, 4702
\ctab .......................................................... 4889, 5448
\ctabtext ..................................................... 4893, 5458
\D .............................................................. 4957, 4989, 4990, 5347, 5352
\dcoli .......................................................... 4958, 4990
\dcoli .......................................................... 4959, 4990
\dcoliv ........................................................ 4960, 4991
\dcolix .......................................................... 4965, 4992
\dcolv .......................................................... 4961, 4991
\dcolvi .......................................................... 4962, 4991
\dcolm .......................................................... 4963, 4992
\dcolm .......................................................... 4964, 4992
\dcolm .......................................................... 4966, 4992
\dcolm .......................................................... 4967, 4993
\dcolm .......................................................... 4968, 4993
\dcolm .......................................................... 4969, 4993
\dcolm .......................................................... 4970, 4994
\dcolm .......................................................... 4975, 4995
\dcolm .......................................................... 4971, 4994
\dcolm .......................................................... 4972, 4994
\dcolm .......................................................... 4973, 4995
\dcolm .......................................................... 4974, 4995
\dcolm .......................................................... 4976, 4995
\dcolm .......................................................... 4977, 4996
\dcolm .......................................................... 4978, 4996
\dcolm .......................................................... 4979, 4996
\dcolm .......................................................... 4980, 4997
\dcolm .......................................................... 4985, 4998
\dcolm .......................................................... 4981, 4997
\dcolm .......................................................... 4982, 4997
\dcolm .......................................................... 4983, 4998
\dcolm .......................................................... 4984, 4998
\dcolm .......................................................... 4986, 4998
\DeclareOptionX ............................................... 21–26, 34–42, 49
\default@series ................................................. 21, 3403
Dekker, Dirk-Jan ................................................. 8, 53
\Dendnote ....................................................... 19
\Detokenize ....................................................... 19
\Dfootnote ....................................................... 19
\Dfootnote ....................................................... 20
Index

edarray1 (environment) ........................................ 45, 5538
edarray (environment) ........................................ 45, 5538
\EDTAB .................................................. 5488, 5496
\edtab .................................................. 4896, 5494
\edatabell ............................................... 4897, 5478
\edatleft ................................................ 47, 4898, 5279
\edatright ................................................ 47, 4899, 5287
\edbeforetab ............................................. 47, 235, 4894, 5394, 5430, 5448
\edfilldimen .............................................. 5301, 5311, 5312, 5314, 5337
\edfontinfo ............................................. 1041, 1044, 1048
\EDINDEX ................................................. 5017
\edindex ................................................. 48
4542, 5017, 5098, 5103, 5109, 5114, 5126, 5135, 5154, 517, 5182, 5188, 5203, 5220
\edindexlab .............................................. 44, 4463, 4469, 4472, 4490, 4508, 4677, 4681, 4686, 4688, 4718
\EDLABEL .................................................. 5015
\edlabel ................................................. 39, 959, 3885, 4469, 5015, 5116, 5132, 5134, 5153, 5171, 5187, 5202, 5219, 5353, 5360, 5365, 5373, 5378, 5386
\edlineref ................................................ 39, 3986
\edmakelabel ............................................ 40, 4087
\edpageref .............................................. 39, 3983
\edrowfill .............................................. 46, 4902, 5117, 5210, 5335
\EDTAB .................................................. 5484, 5520
\edtabcolsep 46, 5074, 5149, 5168, 5183, 5199, 5215, 5232, 5312, 5368, 5381, 5390, 5506
\EDTABINDENT ........................................... 5501, 5514
\edtabindent ............................................. 5342, 5346, 5351, 5362, 5375, 5388, 5510
\EDTABtext ............................................... 5528
edtabularc (environment) ................................ 45, 5542
edtabularl (environment) ................................ 45, 5542
edtabularr (environment) ................................ 45, 5542
\EDTEXT ................................................ 5011
\edtext ................................................ 48, 4890, 5298
\edvertdots .............................................. 48, 4890, 5296
\edvertline ............................................. 48, 4900, 5296
\Eendnote ................................................. 20
\Efootnote ................................................. 19
\eled@chapter .......................................... 5923, 5927, 5975, 5979, 6022
\eled@section .......................................... 5936, 5940, 5988, 5992, 6022
\eled@sectioning@out 271, 317, 5914, 5926, 5939, 5952, 5964, 5978, 5991, 6004, 6017
\eled@sectioningR@out .................................. 5922, 5935, 5948, 5960, 5974, 5987, 6000, 6013
\eled@sections@0 ........................................ 268, 1334, 1360
\eled@subsection ........................................ 5949, 5953, 6001, 6005, 6022
\eled@subsubsection .................................... 5961, 5965, 6014, 6018, 6022
\eledchapter ............................................. 5919
\eledchapter* ........................................... 5919
\eledmac@error .......................................... 80, 82, 84, 86,
88, 100, 125, 128, 131, 134, 136, 139, 198, 200, 203, 205, 207, 209, 212, 215, 218
\eledmac@warning 79, 103, 105, 107, 109, 111, 113, 115, 118, 121, 123, 141, 143, 145,
147, 150, 152, 154, 156, 159, 162, 166, 168, 173, 175, 180, 182, 186, 189, 192, 195
Index

\eledmac@xindy@out ........................................ 43, 44, 47, 4486, 4496, 4712
\eledmacmarkplocrefdepth .................................. 46, 4489, 4499, 4715
\eledsection ............................................... 5787, 5919
\eledsection* ............................................... 5919
\eledsubsection ............................................ 5919
\eledsubsection* ........................................... 5919
\eledsubsubsection .......................................... 5919
\eledsubsubsection* .......................................... 5919
\emph ....................................................... 4541
\empty ...................................................... 1012, 1039, 1063, 1067, 1073, 1150, 1167, 1228, 1447, 1510, 1526, 1648–1650, 1661, 1693, 3892, 3905, 4792
\enablel@dtabfeet ........................................... 5402, 5418, 5434, 5446, 5456, 5464, 5542, 5548
\end@lemmas .................................................. 951, 1012, 1013
\endashchar .................................................. 33, 1827, 1976, 3155
\endgraf ...................................................... 1255, 1305, 1309
\endline@num .................................................. 523, 810, 816
\endlock ...................................................... 17, 927, 958, 4848, 4866, 4875
\endminipage .................................................. 4384
\endnumbering ................................................ 12, 139, 248, 292, 326, 345
\endpage@num .................................................. 523, 809, 816
\endprint ..................................................... 3025, 3173, 3184
\endquotepage ................................................ 5555
\endquote ..................................................... 5555
\endstanzaextra ............................................. 37, 4839
\endsub ....................................................... 17, 903, 957
\endsubline@num ............................................. 523, 811, 817
\enlargespace ................................................ 6111, 6119
\enspace .................................................... 3053

environments:
edarrayc ............................................. 45, 5538
edarrayl ...................................................... 45, 5538
edarrayr ...................................................... 45, 5538
edtabularc .................................................... 45, 5542
edtabularl .................................................... 45, 5542
edtabularr .................................................... 45, 5542
ledgroup ..................................................... 38, 4411
ledgroupsized ............................................... 38, 4425
minipage ..................................................... 38

Euclid .......................................................... 10
\ExecuteOptionsX ............................................ 52, 53
\expandonce ................................................ 827, 1744, 1755, 2426, 2543, 3275, 3292, 3345, 3364, 3365, 4171, 4175, 4186, 4190, 4201, 4205, 4601
\extensionchars ............................................. 51, 232, 254, 333, 570, 572
\extractendline@ ........................................... 3603, 3606
\extractendsubline@ ........................................ 3604, 3606
\extractline@ .............................................. 3601, 3606, 3609
\extractsubline@ ............................................ 3602, 3606, 3609

F
\f@encoding .................................................. 1048
Index

\f@family .................................................. 1048
\f@series .................................................. 1048
\f@shape ................................................... 1048
\f@x@f@ck .................................................. 1330, 1553, 1559
Fairbairns, Robin ........................................ 42
\falseverse ............................................... 4839
\first@linenum@out@false ................................. 834, 841
\first@linenum@out@true ................................ 834
\firstlinenum .............................................. 16, 424
\firstseriesX@ ............................................ 2942, 2942, 2947, 2950, 2953, 2955, 2969
\firstsublinenum ......................................... 16, 424
\firstXseries@ ............................................ 2921, 2921, 2926, 2929, 2932, 2934, 2936, 2957, 3755
\fix@page .................................................. 578, 624
\flag@end .................................................. 878, 1001, 1018, 1019, 1029
\flag@start ............................................... 878, 1000, 1001, 1019
\flagstanza ............................................... 37, 4879
\floatingpenalty ......................................... 1773
\flush@notes .............................................. 1265, 1691, 4336
\flush@notesR ............................................. 4334
\fp@pos ..................................................... 43, 2901
Folkerts, Menso ........................................... 10
\fontencoding ............................................. 1705
\fontfamily ............................................... 1705
\fontseries ............................................... 1705
\fontshape ............................................... 1705
\footfootmarkA .......................................... 42
\footfudgefiddle ......................................... 52, 2090, 2095, 2760
\footins ................................................... 3717, 3724, 3728, 3781, 3824
\fontnormal .............................................. 2096, 3307, 3796
\fontnormalX ............................................. 42, 2559, 2908, 3350
\footnote@luatexpardir .................................. 1792, 1801, 2477, 3343
\footnote@luatextextdir .................................. 1791, 1800, 1832, 2476, 3342
\footnoteA ................................................ 42
\footnoteB ................................................ 42
\footnoteC ................................................ 42
\footnoteD ................................................ 42
\footnoteE ................................................ 42
\footnotelang@lua ......................................... 1721, 3264, 3281
\footnotelang@poly ....................................... 1735, 3268, 3285
\footnoteloptions@ ....................................... 1708, 3270, 3277, 3286, 3295
\footnoterule ............................................. 2006, 2512, 3727, 4399
\footnotesize ............................................. 3585, 4140, 4141
\footnotexnomk .......................................... 3338
\footnoteZ ................................................ 42
\footparagraph .......................................... 24, 2068
\footparagraphX .......................................... 42, 2732
\footsplitskips .......................................... 1763, 1770, 2120, 2135, 2208, 2338, 2456, 2614, 2688, 2790, 2804
\foottwocol .............................................. 24, 2239
\foottwocolX ............................................. 42, 2659
\foottwocol .............................................. 24, 2314
Index

\ifreportnoidxfile ........................................... 4549
\ifrightnoteup ............................................. 4163, 4260
\ifseriesbefore ............................................. 2932, 2953, 3418
\ifshowindexmark ........................................... 4562
\ifsidepstartnum ........................................... 1235, 1605
\ifstrempty ................................................ 1206, 1274, 3433, 3435, 3450, 3452, 3464, 3477,
   4653, 6023, 6025, 6036, 6043, 6044, 6054, 6061, 6062, 6072, 6079, 6080, 6084
\ifStrEq ..................................................... 567, 853, 863, 1122, 1321, 1337, 2218, 2926, 2947,
   3627, 3736, 3739, 4063, 4080, 4316, 4319, 4700, 4706, 6107, 6116, 6129, 6133
\ifStrEqCase ................................................ 875
\ifstrequal ............................................... 1710, 1722, 1736, 3426, 3446
\ifsublines ................................................ 482, 514,
   614, 644, 649, 655, 670, 688, 697, 711, 733, 815, 817, 1399, 1436, 1496, 3924, 3948
\ifSubStr ................................................... 3048, 3050, 3667, 3669, 4606, 4616, 4672, 4696
\iftoggle ....................................................... 1822, 1990, 2103, 2159, 2280, 2350,
   2494, 2768, 2866, 2879, 2890, 2896, 3030, 3605, 3611, 3616, 3621, 3679, 3685
\iftoggle ....................................................... 4529, 4671
\ifvbox ....................................................... 1258, 3697, 3783
\ifvmode ..................................................... 3897, 3910
\ifvoid ....................................................... 2973, 2986, 3002, 3005, 3011, 3717, 3757, 3773, 3781,
   3798, 3801, 3806, 3814, 3815, 3824, 3829, 3834, 4340, 4363, 4387, 4419, 4446
\ifwidthliketwocolumns ................................. 263, 338, 3026, 3169
\ifxendinsertsep ........................................... 4, 4481
\ifxindy ....................................................... 4, 4485, 4695
\ifxindyhyperref ........................................... 75, 76, 4608, 4609, 4612
\indexentry .................................................. 4619, 4622, 4626
\indentl@wrindexentry ...................................... 75, 76
\indtl@wrindexentry ......................................... 245
\inputIfFileExists ......................................... 262, 337, 5555
\input@lemmaseparator ...................................... 28
\input@number ............................................... 27
\inbox .......................................................... 270, 556
\insert ...................................................... 1760,
   2116, 2266, 2336, 2453, 2612, 2686, 2786, 2987, 3006, 3774, 3781, 3783, 3802
\insert@count ............................................... 787, 788, 892,
   894, 997, 1712, 1724, 1726, 1739, 1742, 1745, 2428, 2545, 3294, 4177, 4192, 4207
\insert@countR .............................................. 883, 885, 996, 1716, 1730, 1732, 1750, 1753, 1756, 3278, 4173, 4188, 4203
\insertparagraph ........................................... 1350, 4770
\insertparagraph ........................................... 1325
\insertparagraph ........................................... 1323
\insertparagraph ........................................... 4769
\insert@list ................................................ 302, 518, 551, 796, 1650, 1654
\insertparagraph ........................................... 2155, 2214, 2817
\insertparagraph ........................................... 1227, 1645, 1648, 1658, 1693,
   1694, 1711, 1723, 1725, 1738, 1741, 1744, 2427, 2544, 3293, 4176, 4191, 4206
\insertparagraph ........................................... 1715, 1729, 1731, 1749, 1752, 1755, 3276, 4172, 4187, 4202
\insertparagraph ........................................... 4869
\insertparagraph ........................................... 4842
Index

281

\l@dpagingtrue .................................................. 236
\l@dpairingfalse .................................................. 236
\l@dpairingtrue .................................................. 236
\l@dparsedendline ................................................ 1857, 3965, 3971, 4474
\l@dparsedendpage ............................................. 1857, 3965, 3971, 4474
\l@dparsedendsub ................................................. 1857, 3965, 3971
\l@dparsedstartline ............................................. 1857, 3962, 3968, 4473
\l@dparsedstartpage ............................................. 1857, 3962, 3968, 4473
\l@dparsedstartsub ............................................. 1857, 3962, 3968
\l@parsefeetext .................................................. 1857
\l@dprefirstcolumnfalse ......................................... 236
\l@dprefirstcolumntrue .......................................... 236
\l@dprefirstpagesfalse .......................................... 236
\l@dprefirstpagestrue ............................................ 236
\l@dpush@begins .................................................. 236
\l@dpush@begins .................................................. 236
\l@dpushingfalse .................................................. 236
\l@dpushingtrue .................................................. 236
\l@drd@ta .......................................................... 4751, 4755
\l@drd@ta .......................................................... 1352, 1379, 1534, 1613, 1621, 5701, 5704, 5717
\l@dref@undefined ................................................ 3983, 3986, 3992, 3995, 3998
\l@drestorefills ................................................ 5115, 5399, 5415, 5435, 5443, 5453, 5461
\l@drestoredtext ................................................ 5094, 5548
\l@drestoredtext ................................................ 5105, 5551
\l@drp@box ........................................................ 1389, 4133
\l@drsn@te ........................................................ 4144, 4163
\l@dsn@te .......................................................... 5006, 5047, 5059
\l@dskipnumberfalse ............................................. 930, 1494
\l@dskipnumbertrue .............................................. 930, 1480
\l@dskipversenumberfalse ....................................... 1250
\l@dskipversenumbertrue ......................................... 1482
\l@dstartendokfalse ............................................. 5322, 5326, 5330
\l@dstartendoktrue ............................................. 5320
\l@dtabladdcols ................................................ 5306, 5336
\l@dtabnoexpands ................................................ 964, 4887
\l@dunboxxmpfoot ................................................. 4388, 4396, 4420, 4447
\l@dunboxxmpfoot ................................................. 1312, 1351, 1367, 1370
\l@dzerosize ...................................................... 1245, 1254, 1258
\l@dimakeidxtrue ............................................... 70, 74
\l@dinserttoolstrue ............................................. 73
\l@dluatextextdir@L ............................................ 1211, 1348
\l@dprev@npb .................................................... 280, 854, 865, 6094, 6101, 6102, 6108, 6117
\l@dprev@qb ..................................................... 279, 6093, 6099, 6100, 6105
Lück, Uwe .......................................................... 8
\l@label .......................................................... 40
\l@label@refs .................................................. 3893, 3895, 3901, 3906, 3908, 3914, 3922, 3925, 3927, 3929
\l@label@startfalse ............................................. 1194
\l@label@starttrue .............................................. 1194
\l@label@re@ol@ist ............................................. 3881, 3905, 3908, 3948
\l@label@re@ol@ist@R .......................................... 3892, 3895
\l@label@re@sparsesubline ..................................... 3919
\l@label@re@sparsesubline ..................................... 3919
\l@language@name ................................................ 1744, 1755
<table>
<thead>
<tr>
<th>Warning</th>
<th>Page Numbers</th>
</tr>
</thead>
<tbody>
<tr>
<td>\ledwarn@AddfootinsXobsolete</td>
<td>155, 2997</td>
</tr>
<tr>
<td>\ledwarn@AddfootinsXobsolete</td>
<td>155</td>
</tr>
<tr>
<td>\ledwarn@AppLabelDoutEdtext</td>
<td>142, 3975</td>
</tr>
<tr>
<td>\ledwarn@BadAction</td>
<td>140, 1486</td>
</tr>
<tr>
<td>\ledwarn@BadAdvancelineLine</td>
<td>114, 664</td>
</tr>
<tr>
<td>\ledwarn@BadAdvancelineSubline</td>
<td>114, 658</td>
</tr>
<tr>
<td>\ledwarn@BadLineation</td>
<td>102, 380</td>
</tr>
<tr>
<td>\ledwarn@BadLinenummargin</td>
<td>102, 408</td>
</tr>
<tr>
<td>\ledwarn@BadLockdisp</td>
<td>102, 435</td>
</tr>
<tr>
<td>\ledwarn@BadSetline</td>
<td>120, 915</td>
</tr>
<tr>
<td>\ledwarn@BadSetlinenum</td>
<td>120, 922</td>
</tr>
<tr>
<td>\ledwarn@BadSidenotemargin</td>
<td>151, 4125</td>
</tr>
<tr>
<td>\ledwarn@BadSublockdisp</td>
<td>102, 461</td>
</tr>
<tr>
<td>\ledwarn@DuplicateLabel</td>
<td>142, 3937, 3956</td>
</tr>
<tr>
<td>\ledwarn@LineFileObsolete</td>
<td>112</td>
</tr>
<tr>
<td>\ledwarn@NoIndexFile</td>
<td>153, 4550</td>
</tr>
<tr>
<td>\ledwarn@NoLineFile</td>
<td>110, 561</td>
</tr>
<tr>
<td>\ledwarn@NoMarginpars</td>
<td>149, 4092</td>
</tr>
<tr>
<td>\ledwarn@Obsolete</td>
<td>112, 570, 572</td>
</tr>
<tr>
<td>\ledwarn@RefUndefined</td>
<td>142, 4000</td>
</tr>
<tr>
<td>\ledwarn@SeriesStillExist</td>
<td>161, 3203</td>
</tr>
<tr>
<td>\ledchapter</td>
<td>5555</td>
</tr>
<tr>
<td>\ledchapter*</td>
<td>5555</td>
</tr>
<tr>
<td>\ledfinalfalse</td>
<td>37</td>
</tr>
<tr>
<td>\ledfinaltrue</td>
<td>36</td>
</tr>
<tr>
<td>\ledfootinsdim</td>
<td>2036, 3251, 3321</td>
</tr>
<tr>
<td>ledgroup (environment)</td>
<td>38, 4411</td>
</tr>
<tr>
<td>ledgroupsize (environment)</td>
<td>38, 4425</td>
</tr>
<tr>
<td>\ledinnernote</td>
<td>41, 4143</td>
</tr>
<tr>
<td>\lednote</td>
<td>4490, 4500, 4527</td>
</tr>
<tr>
<td>\lednotehyperpage</td>
<td>4527</td>
</tr>
<tr>
<td>\lednotemark</td>
<td>4527</td>
</tr>
<tr>
<td>\leleftnote</td>
<td>41, 4143, 4928, 4931, 4936</td>
</tr>
<tr>
<td>\ledlinenum</td>
<td>473</td>
</tr>
<tr>
<td>\ledlfill</td>
<td>1346, 1391, 4429, 4433</td>
</tr>
<tr>
<td>\lednotefontsetup</td>
<td>42, 4136, 4250</td>
</tr>
<tr>
<td>\lednoteqsep</td>
<td>42, 1387, 4136</td>
</tr>
<tr>
<td>\lednotewidth</td>
<td>41, 4136, 4250</td>
</tr>
<tr>
<td>\lednopb</td>
<td>50, 6095</td>
</tr>
<tr>
<td>\lednophysversetric</td>
<td>39, 50</td>
</tr>
<tr>
<td>\lednophysnum</td>
<td>6095, 6135</td>
</tr>
<tr>
<td>\ledouternote</td>
<td>41, 4154</td>
</tr>
<tr>
<td>\ledouterote</td>
<td>4143</td>
</tr>
<tr>
<td>\ledpbo</td>
<td>50, 6095</td>
</tr>
<tr>
<td>\ledpbnnum</td>
<td>6095, 6131</td>
</tr>
<tr>
<td>\ledpbrsetting</td>
<td>50, 6103</td>
</tr>
<tr>
<td>\ledplinenumtrue</td>
<td>3587</td>
</tr>
<tr>
<td>\ledrightnote</td>
<td>41, 4143, 4927, 4930, 4935</td>
</tr>
<tr>
<td>\ledrfill</td>
<td>1352, 1391, 4430, 4437</td>
</tr>
<tr>
<td>\ledrsnotefontsetup</td>
<td>42, 4136, 4258</td>
</tr>
</tbody>
</table>
\linenumsep .......................................................... 17, 473, 1635, 1640, 4138, 4139
\lineref .............................................................. 3986
\linewidth .......................................................... 1341
\list@clear ........................................................... 498, 550–553, 1227
\list@clearing@reg .................................................. 538, 549
\list@create ........................................................ 497, 518–521, 818, 823, 951, 1645, 3881
\listbreak ............................................................ 1127
\listctsgadd .......................................................... 6040, 6058, 6076, 6088
\liststeadd .......................................................... 3407, 3408, 3414, 3415
\liststgadd ........................................................... 4224, 4226, 4242, 4244, 4247
\listxadd ............................................................ 636, 3400, 6099–6102
\lock@disp ............................................................ 429, 1577, 1581, 1585
\lock@off .............................................................. 704, 705, 731, 928
\lock@on ............................................................... 702, 927
\lockdir ............................................................... 17, 107, 420
Lorch, Richard ....................................................... 10
\Lpack ................................................................. 4640
\ltab ................................................................. 4890, 5430, 5538
\ltabtext .............................................................. 4892, 5440, 5542
\luatexpardir ........................................................ 1725, 1731, 1792, 1801, 2477, 3343
\luatextextdir ....................................................... 479, 1211, 1348, 1723, 1729, 1791, 1800, 1956, 2476, 3342
Luecking, Dan ......................................................... 56
\makecolfloats ....................................................... 3602, 3711
\makecolintro ......................................................... 3692
\makecoltext ......................................................... 3692, 3712
\m@makeinextrafeet ................................................ 3788
\m@doextrafeet ....................................................... 3787
\m@mmf@check ......................................................... 2394, 2408, 2438
\m@mmf@prepare ..................................................... 2391, 2403, 2412, 2442, 3345
\M@sect .............................................................. 5725
\m@th ................................................................. 5299
\makeboxofhboxes ................................................... 2168, 2198, 2263, 2826, 2854
\makememindexhook ................................................ 4543
\managestanza@modulo ............................................. 4798, 4829
\marginparwidth ..................................................... 4136, 4137
\mathchardef .......................................................... 4793
\mathdepth ........................................................... 3713
\maxdimen ........................................................... 2121, 2136, 2791, 2805
\maxnotesX .......................................................... 32
\maxXnotes .......................................................... 31
Mayer, Gyula .......................................................... 10
\measurebody ........................................................ 5398, 5404, 5434, 5452
\measurecell ........................................................ 5039, 5065
\measurecol ........................................................ 5063, 5409
\measureextrafeet ................................................... 5414, 5420, 5442, 5460
\measurecell ......................................................... 5051, 5070
Index

\newif ............................................. 4-18, 66, 69, 71, 77, 78, 235-240, 242-244, 350, 351, 514, 527, 771, 834, 878, 930, 931, 1036, 1057, 1097, 1190, 1200, 1202, 1289, 1311, 1606, 1631, 1850-1856, 1873, 3015, 3169, 3586, 3715, 3808, 4163, 4265, 4453, 4454, 4475, 4768, 4769, 5318, 5693
\newlength ........................................... 3253, 3255, 3237, 3229
\newlinechar ........................................... 3358
\newread ................................................ 536
\newrobustcmd ........................................ 3978
\newseries ........................................... 3197, 3402, 3403, 3198, 3202
\newseries@eledpar .................................... 3205, 3206
\newtoggle ............................................. 2037, 2041, 3209, 3210, 3227, 3228, 3231-3235, 3237, 3240, 3252, 3310, 3322, 3378-3380, 3396, 3590-3593, 4044, 4045, 4049, 4050
\newverse ............................................. 4839
\newwrite ............................................. 44, 833, 3014, 5914
\NEXT ................................................. 5035, 5040, 5043, 5048, 5049, 5052, 5055, 5056, 5061, 5064, 5066, 5067, 5069, 5071, 5072, 5077, 5236, 5239, 5241, 5242, 5244, 5246, 5247, 5250, 5252, 5253, 5255, 5257, 5258, 5261, 5263, 5264, 5266, 5268, 5269, 5274, 5276, 5277, 5472, 5475, 5476, 5481, 5485, 5486, 5502, 5508, 5509, 5077, 5137, 5139, 5150, 5151, 5156, 5158, 5169, 5170, 5173, 5175, 5184, 5185, 5189, 5191, 5200, 5201, 5204, 5206, 5216, 5217, 5221, 5223, 5233, 5234, 5490, 5492, 5493
\next@absline .......................................... 864, 865
\next@action .......................................... 141, 546, 1418, 1426, 1427, 1433, 1434, 1442, 1451
\next@actionline ...................................... 543, 545, 1417, 1425, 1448, 1450
\next@insert .......................................... 1228, 1649, 1652, 1654, 1657, 1661
\next@page$\num ..................................... 286, 589, 591, 635, 685
\no@expands .......................................... 955, 993, 1053
\noalign ............................................... 2231
\nocritical@true ..................................... 23
\noeledsec ............................................ 49, 186, 5915
\noend@true ........................................... 32
\noendnotes ............................................ 195, 3191
\nofamiliar@true .................................... 24
\noindent ............................................. 1195, 1206, 1274, 1281, 1306, 1603, 1796, 1809, 1814, 2005, 2006, 2008, 2114, 2122, 2126, 2137, 2171, 2201, 2286, 2309, 2356, 2383, 2792, 2806, 2829, 2857
\nointerlineskip ...................................... 2907, 2909, 2915, 2917
\noledgroup@true ..................................... 25
\nolemmaseparator ..................................... 28, 3594
\nomk@ ................................................ 3593
\nonbreakableafternumber ................................ 27
\nonum@ ................................................. 3591
\nonumberinfootnote .................................. 26
\noquotations@true ................................... 34
\normal@footnotemark@X ................................ 2445, 2562
\normal@page$\break ................................ 278, 636, 855, 866, 6092, 6110, 6118
\normal@pars .......................................... 295, 1205, 1229, 1273, 1309, 1795, 1804, 2271, 2341, 2622, 2696
\normal@bfnote@X .................................... 2540, 2554
\normal@body@footnotemark@X .......................... 2450, 2563
Index

\texttt{\textbackslash one@line} .......................... 1189, 1318, 1319, 1346, 1351, 1367, 1370
\texttt{\textbackslash onlypstartinfnote} ........................................ 26
\texttt{\textbackslash openout} .................................................. 43, 271, 842, 849, 3016

\texttt{P}
\texttt{\textbackslash p@pstart} .................................................. 1243
\texttt{\textbackslash PackageError} .............................................. 80
\texttt{\textbackslash PackageWarning} ........................................... 79
\texttt{\textbackslash page@action} .............................................. 590, 683, 802
\texttt{\textbackslash page@num} .................................................. 168, 175, 182, 523, 541,
  633, 809, 814, 1427, 1543, 1618, 2215, 2224, 4147, 4155, 4220, 4238, 4293, 6128
\texttt{\textbackslash page@numR} .............................................. 166, 173, 180, 4215, 4233
\texttt{\textbackslash page@start} ............................................... 901
\texttt{\textbackslash pagebreak} ................................................ 6105
\texttt{\textbackslash pagelinesep} ............................................ 44, 4463, 4472–4474
\texttt{\textbackslash pageno} ................................................ 143, 145, 147, 3688
\texttt{\textbackslash pageparbreak} ........................................... 52, 1603
\texttt{\textbackslash pgereff} ................................................... 40
\texttt{\textbackslash pairs} .................................................. 6029, 6032, 6047, 6051, 6065, 6069
\texttt{\textbackslash paperwidth} .............................................. 1366, 1369
\texttt{\textbackslash para@line} ................................................ 1189, 1256, 1669, 1670, 1673, 1677
\texttt{\textbackslash para@footgroup} ....................................... 2074, 2164
\texttt{\textbackslash para@footgroupX} .................................... 2738, 2822
\texttt{\textbackslash para@footsetup} ....................................... 2080, 2091
\texttt{\textbackslash para@footsetupX} .................................... 2745, 2756
\texttt{\textbackslash para@vfootnote} ....................................... 2072, 2115
\texttt{\textbackslash para@vfootnoteX} .................................... 2736, 2785
\texttt{\textbackslash parafootfmt} ........................................... 2073, 2154
\texttt{\textbackslash parafootfmtX} ........................................... 2737, 2813
\texttt{\textbackslash parafootfmtsep} ...................................... 3352, 3599
\texttt{\textbackslash parafootsep} .......................................... 31
\texttt{\textbackslash parafootstart} ........................................... 2071, 2099
\texttt{\textbackslash parafootstartX} ...................................... 2735, 2765
\texttt{\textbackslash parapparatus@false} .................................. 19
\texttt{\textbackslash parapparatus@true} ................................... 38
\texttt{\textbackslash parindentX} ............................................ 29
\texttt{\textbackslash parledgroup@} ........................................... 2016, 2179, 2293, 2367, 2522, 2639, 2712, 2835
\texttt{\textbackslash parledgroup@beforenotesL} ............................. 4345, 4405
\texttt{\textbackslash parledgroup@beforenotesR} ............................. 4343, 4403
\texttt{\textbackslash parledgroup@series} .................................. 2017, 2180, 2294, 2368, 2523, 2640, 2713, 2836
\texttt{\textbackslash parledgroup@type} ................................... 2018, 2181, 2295, 2369, 2524, 2641, 2714, 2837
\texttt{\textbackslash patchcmd} ................................................ 5620, 5623, 5624, 5627, 5631, 5632, 5727, 5749, 5757, 5767,
  5775, 5784, 5794, 5803, 5812, 5847, 5855, 5863, 5871, 5880, 5888, 5896, 5904
\texttt{\textbackslash pausenumbering} ....................................... 13, 324
\texttt{\textbackslash pend} ..................................................... 13, 88, 131, 134, 1225, 1247, 1307, 1603, 4867, 4875, 5558, 5560, 5566, 5568,
  5574, 5576, 5582, 5584, 5590, 5592, 5598, 5600, 5609, 5612, 5615, 5617, 5630
\texttt{Plato of Tivoli} ...................................................... 10
\texttt{\textbackslash postbodyfootmark} .................................... 2434, 2448
\texttt{\textbackslash postdisplaypenalty} .................................. 1287
\texttt{\textbackslash prebodyfootmark} .................................... 2434, 2446
Index

\pstartnumfalse ................................................................. 1636, 1643
\pstartnumtrue ................................................................. 1269, 1632
\pstartref ................................................................. 39, 3995

\quote ................................................................. 5555
\quote ................................................................. 5555

Q

\RaggedLeft ................................................................. 2166, 2196, 2824, 2852
\RaggedRight ................................................................. 2167, 2197, 2825, 2853
\raggedright ................................................................. 3215, 3315, 4141, 5804, 5806
\raggedX ................................................................. 31
\rav@text ................................................................. 1189, 1231, 1258, 1318
\rbracket ................................................................. 33, 1827, 3244
\read@linelist ................................................................. 536, 838
\ref ................................................................. 5035, 5489, 5496
\rem@inder ................................................................. 1522, 1524–1526
\removeboxes ................................................................. 2169, 2199, 2203, 2827, 2855
\removelastskip ................................................................. 5136, 5155
\RequirePackage ................................................................. 20, 56–58, 60–65
\reserveinserts ................................................................. 59
\resetprevline@ ................................................................. 17, 472, 1538, 1546, 5669, 5715
\resetprevline@ ................................................................. 75, 288, 528, 1262, 1430
\resetprevpage@ ................................................................. 532
\resetprevpage@ ................................................................. 75, 289, 542, 4412
\restore@familiarnotes ................................................................. 4905, 4945
\restore@notes ................................................................. 1538, 1546, 5699, 5715
\restore@sidenotes ................................................................. 4926, 4944
\resumenumbering ................................................................. 15, 324
\right ................................................................. 5282, 5285, 5290, 5293
\rightctab ................................................................. 5359, 5451
\rightlinenum ................................................................. 15, 472, 1538, 1546, 5699, 5715
\rightlinenumR ................................................................. 5700, 5716
\rightltab ................................................................. 5372, 5432
\rightnoteupfalse ................................................................. 42
\rightnoteuptrue ................................................................. 4164
\rightpstartnum ................................................................. 1613, 1621, 1638
\righttrtab ................................................................. 5385, 5397
\rightstartnum ................................................................. 1605
\rigidbalance ................................................................. 2226, 2289, 2312, 2359, 2386, 2635, 2657, 2708, 2730
\rlap ................................................................. 1538, 1546, 1613, 1621, 5698, 5714
\Rlineflag ................................................................. 4032–4034, 4602, 4603, 4631, 4658, 4662
\Robinson, Peter ................................................................. 8
\robustify ................................................................. 3589
\roman ................................................................. 5305
\rtab ................................................................. 4888, 5394, 5540
\rtabtext ................................................................. 4891, 5412, 5544
Index

S
Sacrobosco ................................................. 10
\sameword .................................................. 22, 962, 1085
\sameword\inedtext ........................................ 962, 1147
\sc\n\list .................................................. 1523, 1525
Schöpf, Rainer .............................................. 9
\section\num ................................................ 232, 251,
253, 254, 270, 271, 331–333, 570, 572, 1101, 1103, 1104, 1106, 1158, 1160, 3023
\section\num\R ................................................. 1108, 1110, 1111, 1113, 1175, 1177
\section\mark ................................................. 5684, 6047, 6051
\select\@lemmafont ........................................ 956, 1703
\select\@leemafont ......................................... 33, 1703, 1821, 2158, 2279, 2349, 3055
\series ..................................................... 3197
\series\at\begin ............................................ 34, 3404
\series\at\end ............................................... 34, 3411
\set\@line .................................................. 995, 1038
\set\@line\@action ......................................... 583, 668, 675, 686, 804
\set\command\@series ........................................ 3443, 3462, 3493
\set\ist\two\following\lines ................................ 1873, 1922, 3104
\set\@drp\@rbox ............................................. 4249, 4286, 4301, 4303
\set\@drp\@box ............................................. 4257, 4288, 4296, 4308
\set\@drpr\@box ............................................. 4249
\set\line ................................................... 17, 121, 913, 960, 1262
\set\linenum ................................................ 18, 123, 920
\set\mcell\center ........................................... 5202, 5262
\set\mcell\left .............................................. 5171, 5251
\set\mcell\right ............................................ 5134, 5240
\set\mrow\center ........................................... 5260, 5455
\set\mrow\left ............................................... 5249, 5437
\set\mrow\right ............................................. 5238, 5401
\set\notes\X\position\liketwocolumns\@ ........................................}
\set\notes\X\width\liketwocolumns\@ ........................................}
\set\print\endlines ........................................ 2458,
2507, 2529, 2534, 2646, 2651, 2719, 2724, 2757, 2780, 2831, 2842, 2847, 2859
\set\print\lines ............................................ 3076, 3136
\set\stanzaindents ......................................... 1892, 1958
\set\stanzapenalties ....................................... 34, 4798
\set\stanzavalues .......................................... 4788, 4798, 4799, 4885
\set\cell\center ........................................... 5219, 5267
\set\cell\left ............................................... 5187, 5256
\set\cell\right ............................................. 5153, 5245
\set\toggle ................................................ 1711, 1715, 3425
\set\toggle\@series ......................................... 3423, 3475, 3488
\set\row\center ............................................ 5265, 5463
\set\row\left ............................................... 5254, 5445
\set\row\right ............................................. 5243, 5417
\set\X\notes\position\liketwocolumns\@ ........................................}
Index

\setXnoteswidthliketwocolumns@ ........................................ 1765, 2002, 2023, 2028, 2902, 2111, 2176, 2186, 2191, 2300, 2305, 2374, 2379, 2859 \\
\Setzen ................................................................. 5480, 5489, 5491 \\
\showlemma ......................................................... 51, 223, 954, 1011, 1025 \\
\showwordrank ..................................................... 24, 1164, 1181, 1185 \\
\sidenote@margin .................................................. 4098, 4218, 4236, 4291 \\
\sidenote@marginR .................................................. 4103, 4213, 4231 \\
\sidenotecontent@ 4269, 4274, 4275, 4286, 4288, 4296, 4297, 4301, 4303, 4304, 4308 \\
\sidenotemargin .................................................... 41, 4098 \\
\sidenotemmargin .................................................. 152 \\
\sidenotesep ......................................................... 42, 4267, 4275 \\
\skip .............................................................. 1992, 1997, 2014, 2056, 2057, 2063, 2064, 2078, 2079, 2086, 2087, 2105, 2110, 2177, 2245, 2246, 2251, 2252, 2291, 2320, 2321, 2326, 2327, 2365, 2496, 2501, 2520, 2573, 2574, 2580, 2581, 2591, 2592, 2597, 2598, 2637, 2665, 2666, 2671, 2672, 2710, 2743, 2744, 2751, 2752, 2770, 2775, 2779, 2832, 2927, 2928, 2934, 2935, 2948, 2949, 2955, 2956, 2974, 3724, 3758, 3759, 4343, 4345, 4397, 4403, 4405 \\
\skip@lockoff .................................................... 705, 731 \\
\skipnumbering ..................... 18, 930, 4856, 5558, 5560, 5566, 5568, 5574, 5576, 5582, 5584, 5590, 5592, 5598, 5600, 5609, 5615, 5628, 5629, 5637, 5650, 5659, 5672 \\
\spacefactor ................................................. 2396, 2399, 2407, 2413, 2437, 2443 \\
\spaceskip .......................................................... 1768, 2461, 2615 \\
\splitmaxdepth ................................................ 1775 \\
\splitoff .......................................................... 2226 \\
\splittopskip ................................................. 1315, 1775, 2228, 2287, 2289, 2310, 2312, 2357, 2359, 2384, 2386, 2633, 2635, 2655, 2657, 2706, 2708, 2728, 2730 \\
\spreadmath ...................................................... 46, 5468 \\
\spreadtext ............................................... 46, 5466 \\
\stanzaindent .................................................. 34, 4839 \\
\stanzaindent* .................................................. 34, 4839 \\
\stanzaindentline ................................................ 4779, 4790, 4793, 4795, 4824, 4836, 4845, 4855, 4876 \\
\stanzaindents .................................................. 4822, 4847 \\
\stanzaindentbase ................................................. 34, 4779, 4811, 4825, 4828, 4834, 4879 \\
\startlock ......................................................... 17, 927, 958, 4832 \\
\startstanzahook ................................................ 37, 4839 \\
\startsub .......................................................... 17, 903, 957 \\
\stepl@dcolcount ............................................. 3336, 3923, 3926, 4468, 5313 \\
\step1@dcolcount .................................................. 5001, 5046, 5058, 5143, 5162, 5178, 5194, 5209, 5226, 5272, 5473, 5482, 5503 \\
\StrDel ......................................................... 3405, 3412, 4033, 4034 \\
\StrGobbleLeft .................................................. 4517, 4522 \\
\strip@pt .......................................................... 2097, 2763 \\
\strip@szacnt .................................................. 4788 \\
\StrPosition .................................................. 3419, 3420 \\
\sub@change ........................................................ 599, 695, 803 \\
\sub@action ........................................................ 287, 593, 594, 600, 645, 647, 650, 652
Index

\thempfn .............................................. 4377, 4413, 4440
\thempfootnote ....................................... 4377, 4413, 4440
\Theodosius .............................................. 10
\thepage .............................................. 856, 859, 861, 870, 872, 875, 3901, 3914, 4472
\thepageline .............................................. 4471, 4571, 4583, 4593, 4612, 4626
\thepstart ............................................. 14, 1194, 1306, 1634, 1641, 1847
\thepstartL .............................................. 1844
\thepstartR .............................................. 1842
\thestartpageline ...................................... 4473, 4566, 4578, 4588, 4608, 4619
\thesubline ............................................. 3927
\thinspace ................................................... 3927
\thesubline ............................................. 3927
\this@line@list@version ............................................. 567, 836, 843
\this@absline ........................................... 1152, 1156, 1158–1160, 1169, 1173, 1175–1177
\this@line@list@version ............................................. 567, 836, 843
\thisfootnote ............................................. 2542, 2543
\threecolvfootnote ..................................... 406, 716,
725, 736, 743, 1467, 1475, 2260, 2263, 2289, 2312, 2680, 2683, 2708, 2730, 4123
\threecolfootfmt ....................................... 2242, 2270
\threecolfootfmtX ...................................... 2662, 2691
\threecolfootgroup ..................................... 2243, 2285
\threecolfootgroupX ................................... 2663, 2705
\threecolfootsetup ..................................... 2247, 2257
\threecolfootsetupX .................................... 2667, 2677
\threecolvfootnote ..................................... 2241, 2265
\threecolvfootnoteX .................................... 2661, 2685
\tmp ....................................................... 1876, 1877, 1884
\togglefalse ............................................ 1991, 2104, 2495, 2769, 3070, 3071, 4075, 4085
\toggletrue .............................................. 2038, 2042, 3036, 4064, 4081
\tolerance ................................................ 2274, 2344, 2626, 2699
\twocolfootfmt ......................................... 2317, 2342
\twocolfootfmtX ....................................... 2588, 2618
\twocolfootgroup ....................................... 2318, 2342
\twocolfootgroupX ..................................... 2589, 2632
\twocolfootsetup ....................................... 2322, 2342
\twocolfootsetupX ..................................... 2593, 2603
\twocolvfootnote ....................................... 2316, 2342
\twocolvfootnoteX ..................................... 2587, 2611
\twolines ................................................ 25
\twolines@appref ....................................... 4042
\twolines@appref ....................................... 4042
\twolines@appref ....................................... 41
\twolines@notmoreappref .................................. 41
\twolines@notmoreappref .................................. 41
\twolines@onlyin@same@page ................................ 26, 41
\txtbefore@Xnotes ....................................... 31

U
\unexpanded ............................................. 1195, 1281, 3273, 3290, 5923, 5927, 5936,
5940, 5949, 5953, 5961, 5965, 5975, 5979, 5988, 5992, 6001, 6005, 6014, 6018
\unhbox ................................................... 1312, 2148, 2169,
2171, 2199, 2201, 2208, 2212, 2827, 2829, 2855, 2857, 5525, 5526, 5533, 5534
\unkern ................................................... 2397
Index

unless ........................................ 360, 368, 376, 1088, 1100, 1149, 1772, 2968, 2984, 3208, 3254, 3309, 3328, 3354, 3496, 3542, 3559, 3754, 3771, 3813, 3828, 3833, 4325, 4331, 4355, 4362, 4456, 4459, 4906, 4918
unpenalty ........................................ 2151, 2205
unvbox .......................................... 1319, 1780, 2009, 2034, 2132, 2146, 2165, 2195, 2237, 2466, 2515, 2538, 2801, 2823, 2851, 2971, 2989, 3001, 3006, 3703, 3723, 3728, 3747, 3774, 3781, 3783, 3802, 3846, 4394, 4409
unvX ............................................. 2123, 2138, 2145, 2793, 2807
usingcritext .................................... 5546
usingedtext ...................................... 5546

V

\valign ........................................... 2229
\value ........................................... 4806, 4818, 4823, 5310
\Varanana ...................................... 10
\variab .......................................... 5079, 5400, 5416, 5436, 5444, 5454, 5462, 5495
\vbadness ........................................ 1314, 2228
\vbox ............................................ 1231, 1779, 2121, 2131, 2136, 2146, 2465, 2791, 2800, 2805, 2908, 2916, 2970, 3000, 3700, 3720, 3746, 3781, 3785, 4253, 4255, 4261, 4263, 4372, 5281, 5284, 5289, 5292, 5296, 5298, 5299, 5342, 5346, 5351, 5362, 5375, 5388
\vfill ............................................ 3724
\vfill ............................................ 2423
\vfil ............................................. 2229, 3875, 5282, 5285, 5290, 5293, 5296, 5299
\vfil ............................................. 3724
\vfil ............................................. 2423
\vfil ............................................. 4201, 4205, 4211
\vfil ............................................. 4171, 4175, 4211
\vfil ............................................. 4186, 4190, 4211
\vnumfootnoteX ................................ 2552, 2565
\vrule .......................................... 5281, 5284, 5289, 5292, 5296
\vsize ........................................... 2036
\vsplit .......................................... 1318, 2236, 3846

W

\wd ................................................ 1306, 1346, 2125, 2140, 2795, 2809, 5007, 5008, 5131, 5355, 5364, 5367, 5377, 5380, 5389, 5523, 5524, 5531, 5532
Whitney, Ron .................................... 9
\widowpenalty ................................... 1287, 1675
\widthliketwocolumnstrue ...................... 41
\WithSuffix .................................... 4814, 5564, 5580, 5596, 5613, 5971, 5984, 5997, 6010
\wrapversedcrossref ......................... 3978, 3983, 3986, 3992, 3995
Wujastyk, Dominik ................................ 7, 9

X

\x@lemma ........................................ 1013–1015
\xcritext ....................................... 5011, 5124
\xedindex ....................................... 5017, 5103, 5114, 5126
\edlabel ......................................... 5015, 5132
\edtext ......................................... 5011, 5125
\Kendafterlemmaseparator .................... 28
\Kendafternote ................................ 32
Index 297

\Xendbeforelemmaseparator ........................................ 28
\Xendplaceoflemmaseparator ........................................ 28
\Xendinsertsepfalse .................................................. 3175, 3188
\Xendinsertseptrue .................................................... 3031
\Xendlemmadisablefontselection ..................................... 29
\Xendlemmaseparator .................................................. 28
\Xendmorehantvolines ................................................ 26
\Xendmorehantvolines@prefwithpage ................................ 4048
\Xendmorehantvolines@prefwithpage ................................ 41
\Xendnotefontsizem .................................................. 29
\Xendnotenumfont ..................................................... 29
\Xendparagraph ....................................................... 32
\Xendsep ............................................................... 32
\Xendvlines ............................................................ 26
\Xendvlines@prefwithpage ............................................. 4047
\Xendvlines@prefwithpage ............................................. 41
\Xendvlinesbnotmore .................................................. 26
\Xendvlinesbnotmore@prefwithpage .................................. 41
\Xendvlinesonlyinsamepage@prefwithpage ......................... 41
\Xinspaceindent ....................................................... 29
\Xifixinlist ........................................................... 854, 855, 865, 866, 1334, 1360, 3203, 6105, 6108, 6110, 6117, 6118
\xindytrue ............................................................. 45
\xindyhyperreftrue .................................................... 50
\Xledsetnormalparstuff ............................................... 189, 1788, 2156
\xleftappenditem ..................................................... 505
\Xlemmadisablefontselection ........................................ 29
\Xlineref .............................................................. 39, 3986, 4070, 4074, 4084, 4472
\Xnotefontsizem ..................................................... 29
\Xnotenumfont ......................................................... 29
\Xnoteswidthliketwocolumns ........................................ 32
\Xpageref ............................................................... 39, 3983, 4074, 4084
\Xparindent ............................................................ 39, 3995
\Xpretat ................................................................. 39
\Xprat ................................................................. 40, 4020
\Xragged ................................................................. 31
\Xrightappenditem ..................................................... 499, 684, 685, 687, 694, 696, 698, 700, 710, 712, 721, 732, 734, 741, 754, 755, 757, 758, 764, 765, 767, 768, 781, 782, 784, 785, 796, 813, 827, 1133, 1136, 1138, 1711, 1715, 1723, 1725, 1729, 1731, 1738, 1741, 1744, 1749, 1752, 1755, 2426, 2543, 3270, 3287, 3946, 4171, 4175, 4186, 4190, 4201, 4205
\Xspacekip .............................................................. 1768, 2461, 2615
\Xsublineref ........................................................... 39, 3992, 4074, 4084
\Xxref ................................................................. 39, 3992, 4074, 4084
\zskip ................................................................. 1768, 1776, 2461, 2615
\Zendnote ............................................................. 20
\Zfootnote ............................................................. 19
\z@skip ............................................................... 3882, 3893, 3906, 4023, 4028
Change History

v0.1.0.
  General: First public release ........................................ 1

v0.2.0.
  General: Added tabmac code, and extended indexing ................. 1
  edmac@error: Added edmac@error and replaced error messages .... 57
  ifl@dmemoir: Added ifl@dmemoir for memoir class having been used 56
  morenoexpands: Added \@dtabnoexpands to \no@expands ............ 92

v0.2.1.
  \@lab: Removed page setting from \@lab .......................... 191
  General: Added text about normal labeling ....................... 40
  Bug fixes and match with mempatch v1.8 ........................ 1
  Major changes to insert code when memoir is loaded ............ 186
  ddoxtrafeet: Renamed \doxtrafeet to \@ddoxtrofeet ............ 185
  edlabel: Tweaked edlabel to get correct page numbers .......... 186
  \@ddomakecol: Rewrote \makecol, calling it \@ddomakecol ....... 184
  \@ddodoreinxtrafeet: Renamed \dooreinxtrafeet to \@ddodoreinxtrafeet 186
  \@ddofootinsert: Renamed \dofootinsert as \@ddofootinsert ... 184
  \m@makecolintro, \m@makecolfloats, \m@makecoltext and \m@makecolintro 184
  morenoexpands: Removed some \lets from \no@expands. These were in EDMAC but I feel that they should not have been as they disabled page/line refs in a footnotes .......................................................... 92
  \zz@@@: Minor change to \zz@@@ .................................. 189

v0.2.2.
  General: Improved paragraph footnotes .......................... 1
  New Dekker example ................................................. 1
  Used \providecommand for \@gobblethree to avoid clash with the amsfonts package ................................................. 61
  footfudgefiddle: Added \footfudgefiddle ........................ 134
  \line@list@stuff: Added initial write of page number in \line@list@stuff . 86
  \@parafootsetup: Added \footfudgefiddle to \@parafootsetup .... 134
  \@parafootsetupX: Added \footfudgefiddle to \@parafootsetupX .... 154

v0.3.0.
  \@lab: Replaced \the\line@num by \line@num in \@lab, and similar for sub-lines .................................................. 191
  \@one@reg: Added a bunch of code to \@one for handling \setlinenum 78
  General: Includes edstanza and more ................................ 1
  ledinnum: Added \linenum and \sublinenum to \leftlinenum and \rightlinenum ............................................. 69
  linenumlist: Added \linenumlist mechanism ........................ 61
  \@printendlines: Added \linenum and \sublinenum to \@printendlines 165
  \@printlines: Added \linenum and \sublinenum to \@printlines ... 129
  \@sublinenum: Added \linenumberstyle and \sublinenumberstyle .... 69

v0.3.1.
  General: Not released. Added remarks about the parallel package .... 1
Change History

v0.4.0.
\@iiiminipage: Modified kernel \@iiiminipage and \endminipage to cater for critical footnotes ........................................ 204
General: Added final/draft options ........................................ 55
Added minipage, etc., support ........................................ 1
\ledgroup: Added ledgroup environment ................................ 205
\ledgroupsized: Added ledgroupsized environment .................... 205
\edtext: Added \showlemma to \edtext (and \critext) .................... 93
\footnormal: Added minipage footnote setup to \footnormal ........... 133
\@dfeetendmini: Added \@dfeetbeginmini, \@dfeetendmini and all their supporting code ........................................ 202
\mpnormalfootgroup: Added \mpnormalfootgroup ....................... 131
\mpnormalvfootnote: Added \mpnormalvfootnote ....................... 123
\showlemma: Added \showlemma ........................................ 61
\affixline@num: Changed \affixline@num to cater for sidenotes ....... 113
\l@dgetsidenote@margin: Added \sidenotemargin and \sidenote@margin . 197

v0.4.1.
General: Added code for changing \@doclearpage ....................... 187
Not released. Minor editorial improvements and code tweaks .......... 1
Only change \@footnotetext and \@footnotemark if memoir not used ... 145
\doxtrafeetii: Changed \doxtrafeetii code for easier extensions ....... 185
\edindex: Let eldmac take advantage of memoir’s indexing ............ 209
\print@Xnotes: Added \@opxtrafeetii .................................... 185

v0.5.0.
\@footnotetext: Enabled regular \footnote in numbered text .......... 146
\@rympar: Eliminated \marginpar disturbance ........................... 196
General: Added left and right side notes .............................. 197
Added sidenotes, familiar footnotes in numbered text ................ 1

v0.5.1.
General: Added moveable side note .................................... 197
Fixed right line numbers killed in v0.5 ............................... 1
\affixline@num: Changed \affixline@num to cater for sidenotes ....... 113
\ledgroupsized: Only change \hsize in ledgroupsized environment otherwise page number can be in wrong place ....................... 205
\l@dgetsidenote@margin: Added \sidenotemargin and \sidenote@margin . 197

v0.6.0.
\@lopR: Added \@open, \@openR, \@lopL and \@lopR in anticipation of parallel processing ........................................ 79
\@nl@reg: Added \@nl@reg ........................................ 78
\@nl: Extended \@nl to include the page number ..................... 78
General: Fixed long paragraphs looping .............................. 1
Fixed minor typos ........................................ 1
Prepared for eldpar package ........................................ 1
\fix@page: Added \@last@page@num and \fix@page ...................... 79
\new@line: Extended \new@line to output page numbers ............... 86
\page@start: Made \page@start a no-op ................................ 87
\v1\@dbfnote: Changed \@dbfnote and \v1\@dbfnote as originals could give incorrect markers in the footnotes ............................. 146

v0.7.0.
\@nl@reg: Added \@nl@reg ........................................ 78
\@ref@reg: Added \@ref@reg ........................................ 83
General: eledmac having been available for 2 years, deleted the commented out
original edmac texts .................................................. 1
Maïeul Rouquette new maintainer ........................................ 1
Made macros of all messages ........................................... 57
Replaced all \interfootnotelinepenalty, etc., by just \interfootnotelinepenalty
................................................................. 1
Tidying up for eledpar and ledarab packages .......................... 1
\affixline@num: Added skipnumbering to \affixline@num ............... 113
\do@actions@fixedcode: Added \do@actions@fixedcode ............... 112
\do@actions@next: Added number skipping to \do@actions .......... 111
\do@insidelinehook: Added \do@insidelinehook for use in \do@line .... 109
\endnumbering: Changed \endnumbering for eledpar .................. 64
\f@x@l@cks: Added \ch@cksub@l@ck, \ch@ck@l@ck and \f@x@l@cks .......... 115
\footsplitskips: Added \footsplitskips for use in many footnote styles 123
\get@linelistfile: Added \get@linelistfile .......................... 76
\ifledRcol@: Added \l@dnumpstartsL, \ifl@dpairing and \ifpst@rted for/from eledpar ................................. 62
\initnumbering@reg: Added \initnumbering@reg .................... 63
\l@dcsnotetext@r: Added \l@demptyd@ta ................................ 109
\l@ddofootinsert: Deleted \page@start from \l@ddofootinsert ....... 184
\l@dgeline@margin: Added \l@dgeline@margin ....................... 67
\l@dglock@disp: Added \l@dglock@disp ............................. 68
\l@dgtsidenote@margin: Added \l@dgtsidenote@margin ............. 197
\l@d@nuline: Added \l@dlin@num and \l@duline for use in \do@line .... 109
\l@dunboxmpfoot: Added \l@dunboxmpfoot containing some common code 204
\l@dzeropenalties: Added \l@dzeropenalties .......................... 106
\ledlineum: Added \ledlineum for use by \leftlineum and \rightlineum 69
\line@list@stuff: Deleted \page@start from \line@list@stuff ........ 86
\list@clearing@reg: Added \list@clearing@reg ...................... 76
\n@num: Added \n@num .................................................. 82
\normalbfnoteX: Removed extraneous space from \normalbfnoteX .... 149
\resumenumbering: Changed \resumenumbering for eledpar .......... 65
\setprintendlines: Added \setprintendlines for use by \printendlines . 163
\setprintlines: Added \setprintlines for use by \printlines .......... 127
\skipnumbering: Added \skipnumbering and supports .............. 89
\sublinenum: Added \firstlinenum, \linenumincrement, \firstsublinenum and \linenumincrement ........................................ 68
\sublinenum@p: Using \linenumrep instead of \sublinenum@p ........ 69
Using \sublinenumrep instead of \sublinenum@p .................... 69
\vnumfootnoteX: Removed extraneous space from \vnumfootnoteX .... 149
v0.8.0.
General: Bug on endnotes fixed: in a // text, all endnotes will print and be placed
at the ends of columns () ........................................... 1
v0.8.1.
General: Bug on \edtext ; \critex ; \lemma fixed: we can now use non-switching
commands .............................................................. 1
v0.9.0.
General: No more ledpatch. All patches are now in the main file ................ 1
v0.9.1.
General: Fix some bugs linked to integrating ledpatch on the main file .......... 1
v0.10.0.
General: Corrections to \section and other titles in numbered sections 1
v0.11.0.
General: Makes it possible to add a symbol on each verse’s hanging, as in French
typography. Redefines the command \hangingsymbol to define the character. 1
v0.12.0.
General: For compatibility with eledpar, possibility to use \autopar on the right
side. 1
Possibility to number \pstart. 14
Possibility to number the \pstart with the commands \numberpstarttrue. 1
\ifledRcol: Added \ifledRcol and \ifnumberingR for/from eledpar 62
v0.12.1.
General: Don’t number \pstart of stanza. 1
The numbering of \pstarts restarts on each \beginnumbering. 1
v0.13.0.
General: New stanzaindentrepetition counter to repeat stanza indents every n
verses. 35
New stanzaindentrepetition counter: to repeat stanza indents every n verses. 1
\managestanza@modulo: New stanzaindentrepetition counter to repeat stanza
indents every n verses. 219
v0.13.1.
General: \thePstartL and \thePstartR use now \bfseries and not \bf, which
is deprecated and makes conflicts with memoir class. 1
v0.14.0.
General: Tweaked \edlabel to get correct line number if the command is first
element of a paragraph. 1
\edlabel: Tweaked \edlabel to get correct line number if the command is first
element of a paragraph. 189
v0.15.0.
General: Line numbering can be reset at each \pstart. 65
Possibility to print \pstart number inside. 15
\affixline@num: Line numbering can be disabled. 113
\ifinsert hhingsymbol: New management of hangingsymbol insertion, pre-
venting undesirable insertions. 218
\printlines: Line numbering can be reset at each \pstart. 129
v0.17.0.
\ifinsert hhingsymbol: New new management of hangingsymbol insertion,
preventing undesirable insertions. 218
v1.0.0.
General: \lemma can contain commands. 21
Debug in lineation command 16
New generic commands to customize footnote display. 25, 173
Options nonum and nosep in \Xfootnote. 19
Options of \Xfootnotes. 121
Possibility to have commands in sidenotes. 41
Some compatibility break with eledmac. Change of name: eledmac. 1
\morenoexpands: Change to be compatible with new features 92
v1.0.1.
General: Correction on \numberonlyfirstinline with lineation by \pstart or by
page. 25
<table>
<thead>
<tr>
<th>Version</th>
<th>Changes</th>
</tr>
</thead>
<tbody>
<tr>
<td>v1.1.0</td>
<td>- General: Add \labelpstarttrue.</td>
</tr>
<tr>
<td></td>
<td>- Add \numberonlyfirstintwolines</td>
</tr>
<tr>
<td></td>
<td>- Add \pstartinfotnote and \onlypstartinfotnote</td>
</tr>
<tr>
<td></td>
<td>- New hook to add arbitrary code at the beginning of the notes</td>
</tr>
<tr>
<td></td>
<td>- New options for block of notes</td>
</tr>
<tr>
<td></td>
<td>- New package option: parapparatus</td>
</tr>
<tr>
<td></td>
<td>- New tools to change order of series</td>
</tr>
<tr>
<td></td>
<td>- Sectioning commands.</td>
</tr>
<tr>
<td></td>
<td>\ledfootinsdim: Deprecated \ledfootinsdim</td>
</tr>
<tr>
<td></td>
<td>\preXnotes: New skip \preXnotes@</td>
</tr>
<tr>
<td></td>
<td>\settoggle@series: \settoggle@series switch the global value of the toggle, not only the local value.</td>
</tr>
<tr>
<td>v1.2.0</td>
<td>\endquote: Compatibility of \ledchapter with the memoir class.</td>
</tr>
<tr>
<td></td>
<td>\preXnotes: Debug in familiar footnotes (but introduced by v1.1).</td>
</tr>
<tr>
<td>v1.3.0</td>
<td>\endquote: Quotation and quote environment inside numbered sections.</td>
</tr>
<tr>
<td>v1.4.0</td>
<td>- General: Compatibility with LuaTeX of RTL notes.</td>
</tr>
<tr>
<td></td>
<td>- edtext: Compatibility of edtext (and \critext) with the right-to-left direction (with Polyglossia).</td>
</tr>
<tr>
<td></td>
<td>- \newseries@: Remembers the language of the lemma, in order to create a correct direction for the footnote separator.</td>
</tr>
<tr>
<td></td>
<td>- normalfootfmt: Direction of footnotes with polyglossia.</td>
</tr>
<tr>
<td></td>
<td>- rbracket: Switch the right bracket to a left bracket when the lemma is RTL (needs polyglossia or LuaTeX).</td>
</tr>
<tr>
<td>v1.4.1</td>
<td>- \affixside@note: Remove spurious spaces.</td>
</tr>
<tr>
<td></td>
<td>- \endquote: New option noquotation.</td>
</tr>
<tr>
<td></td>
<td>- \labelrefsparsesubline: Fix bug with \edlabel.</td>
</tr>
<tr>
<td></td>
<td>- vl@dbfnote: Compatibility of standard footnotes with edlemac when these footnotes contain any \edlabel.</td>
</tr>
<tr>
<td>v1.4.2</td>
<td>- General: Debug with some special classes.</td>
</tr>
<tr>
<td>v1.4.3</td>
<td>- General: Add \nbreakableafternumber.</td>
</tr>
<tr>
<td></td>
<td>- Spurious space after familiar footnotes.</td>
</tr>
<tr>
<td>v1.4.4</td>
<td>- General: Label inside familiar footnotes.</td>
</tr>
<tr>
<td>v1.4.5</td>
<td>- General: Bug with komasscript + edlemac + chapter.</td>
</tr>
<tr>
<td>v1.4.6</td>
<td>- General: Bug with memoir class introduced by 1.4.5.</td>
</tr>
<tr>
<td>v1.4.7</td>
<td>\endquote: Compatibility of sectioning commands with \autopar</td>
</tr>
<tr>
<td>v1.4.8</td>
<td>- General: Corrects a bug with parallel texts introduced by 1.1.</td>
</tr>
</tbody>
</table>
v1.4.9.
`\normalbfnoteX`: Allow to redefine `\thefootnoteX` with alph when some packages are loaded. ................................. 149

v1.5.0.
General: Correct indexing when the call is made in critical notes. .......... 206
`\do@insidelinehook`: Added `\do@insidelinehook` for use in `\do@line` ...... 109
`\edindex`: Compatibility with `imakeidx` package, and possibility to use multiple index with `\edindex`. ................................. 209
`\ifFN@bottom`: Use the bottom option of `footmisc` package. ................. 184

v1.5.1.
`\managestanza@modulo`: Correct stanzaindent repetition counter .......... 219
`\normalvfootnoteX`: Fix bug with normal familiar footnotes when mixing RTL and LTR text. .................................................. 147

v1.6.0.
`\falseverse`: Add `\falseverse` macro. ............................................. 220

v1.6.1.
General: Corrects a false hanging verse when a verse is exactly the length of a line. 1
`\AtEveryPstart`: Spurious space in `\pstart`. .................................. 103
`\ifinserthavingsymbol`: Hang verse is now not automatically flush right. ... 218
`\l@dunhbox@line`: Move the call to `\inserthavingsymbol` to allow use `\hfill` inside. ............................................................ 107
`\pend`: Spurious space in `\pend`. ..................................................... 105

v1.7.0.
General: New features for managing page breaks. ................................. 50

v1.8.0.
General: Compatibility with `parledgroup` option of `eledpar` package. .... 1
If `imakeidx` and `hyperref` are loaded, adds `hyperref` in the index. .......... 206
`\endquote`: Correction of sectioning commands in parallel texts. .......... 242
`\get@index@command`: Debug `\get@index@command` and compatibility with `hyperref` package. .................................................. 208
`\newhookcommand@series@reload`: Debug `\beforenotesX` and `\maxhnotesX` which didn’t work. ....................................................... 176
`\prevpage@num`: Correct `\parafootsep` when using with `ledgroup`. ......... 139

v1.8.1.
General: Debug endnotes when more than one series is used (change the position where tools for endnotes are defined). ............................... 161

v1.8.2.
General: Debug compatibility problem with `hebrew` option of `babel` package. 1

v1.8.3.
General: Fixes spurious spaces added by v1.7.0. ................................. 1

v1.8.5.
General: Debug indexing in right column, with `eledpar`. ...................... 206

v1.9.0.
`\doxtrafeet`: Add `\fnpos` to choice the order of footnotes. .................. 185
`\defeteetendmini`: Add `\mpfnpos` to choice the order of footnotes in minipage / `ledgroup`. .................................................... 202

v1.10.0.
General: Add `\pstartref` and `\xpstartref` to refer to a pstart number (extension of `\edlabel`). .................................................... 1
`\endquote`: Correction of sectioning commands in parallel texts. .......... 242
Change History

v1.10.1.
General: Compatibility with cleveref .......................... 1

v1.10.2.
General: Compatibility with cleveref .......................... 1

v1.10.3.
General: Debug of cross-referencing .......................... 1

v1.10.4.
General: Debug of cross-referencing .......................... 1

v1.10.5.
General: Debug of \pausenumbering .......................... 1
Debug of \xxref .......................... 1

v1.10.6.
General: Debug of interaction between \autopar and \pausenumbering .......................... 1

v1.11.0.
General: Add hooks to disable the font selection for lemma in footnote ............. 29

v1.11.1.
General: Correct a bug when a critical note starts with plus or minus ................. 1

v1.12.0.
\@nl@reg: To ensure compatibility with \musixtex, \@l becomes \@l. Consequently, \@l@reg becomes \@nl@reg .................. 77
General: Add \ledinnernote and \ledouternote commands .................. 41
Add \Xendparagraph and related settings .................. 32
Add hyperlink to crossref (needs hyperref package) .................. 38
Compatibility with \musixtex .......................... 1
Debug eledmac sectioning command after using \resumenumbering .......................... 1
Ensure that \imakeidx is loaded before eledmac .................. 206
New hooks: \afterXrule and \afterruleX .................. 31
New options for ragged-paragraph notes .................. 31
New sectioning commands .................. 48
Optional arguments for \pstart and \pend .................. 13
\AtEveryPstart: New optional argument for \pstart to execute code before it .......... 103
\edindex: Use correctly default index when \imakeidx is loaded .................. 209
\endquote: \ledxxx sectioning commands are deprecated and replaced by \eledxxx commands .................. 242
\ifledRcol@: Add \ifledRcol@ for eledpar .................. 62
\initnumbering@reg: \beginnumbering is defined only on eledmac, not on eledpar .................. 63
\l@dcsnote: \l@dlsnote, \l@drsnote and \l@dcsnote defined only one time, in eledmac, including needs for eledpar case .................. 198
\l@dgetsidenote@margin: \sidenotemargin is now directly defined in eledmac to be able to manage eledpar .................. 197
\l@dunhbox@line: \do@line is split in more little commands .................. 108
\newhookcommand@series@reload: Debug \beforenotesX and \maxhnotesX which didn’t work when called after \footparagraphX .................. 176
Debug \beforeXnotes and \maxhXnotes which didn’t work when called after \footparagraph .................. 176
\pend: New optional argument for \pend to execute code after it ............. 105
\stanza: &can have an optional argument: content to be printed after ............. 220
\Stanza can have an optional argument: content to be printed before ............. 220
Add \newverse macro, \falseverse deprecated .................. 220
Change History

v1.12.1.
\texttt{\textbackslash wrap@edcrossref}: Fix spurious spaces. .......................... 192

v1.12.2.
\texttt{\textbackslash dunhbox@line}: Fix a bug with critical notes at the tops of pages (added by v12.0.0) ......................... 107

v1.12.3.
General: Add macros for new messages since v0.7 ......................... 57
Correct bug with side and familiar notes in tabular environments. ........ 1
Debug \texttt{\textbackslash lededxxx} with some paper size ......................... 1
Debug \texttt{\textbackslash ledinnernote} and \texttt{\textbackslash ledouternote} commands in the top of pages. . . . . . . 41
Debug left and right notes (bugs added by 1.12.0) .......................... 1
Underline lemma in \texttt{\textbackslash lededxxx} when using draft mode. ........ 1
\texttt{\textbackslash leledmac@error}: Replaced error messages ................... 57
\texttt{\textbackslash flag@end}: \texttt{\textbackslash flag@start} and \texttt{\textbackslash flag@end} are now defined only one time for edlemac and edelpar .............. 87
\texttt{\textbackslash flag\@start} send a error message when a \texttt{\textbackslash edtext} is done without insert (note) .............................................. 87

v1.12.4.
General: Debug spurious page breaks before \texttt{\textbackslash chapter} (bug added in 1.12.0) . . 1

v1.12.5.
\texttt{\textbackslash edindex@hyperref}: Debug \texttt{\textbackslash edindex} when \texttt{\textbackslash hyperref} is not loaded .......... 213
\texttt{\textbackslash @edindex@hyperref}: Debug \texttt{\textbackslash edindex} in parallel with \texttt{\textbackslash memoir} ................. 246
\texttt{\textbackslash doinsidelinehook}: Added \texttt{\textbackslash dolinehook} and \texttt{\textbackslash doinsidelinehook} .............. 109
\texttt{\textbackslash endnumbering}: Allow to mix parallel columns and normal text when using \texttt{\textbackslash pausenumbering} ........................................ 64
\texttt{\textbackslash @d gobblearg}: \texttt{\textbackslash @d gobblearg} becomes \texttt{\textbackslash @d gob beloptarg} .................... 226
\texttt{\textbackslash @d restoreforedtext}: Debug optional arguments of \texttt{\textbackslash Xfootnote} in tabular context ........................................ 228
\texttt{\textbackslash resumenumbering}: Debug \texttt{\textbackslash resumenumbering} ......................... 65

v1.12.6.
\texttt{\textbackslash @noeledsec}: Add \texttt{\textbackslash noeledsec} macro. ............... 250

v1.12.7.
\texttt{\textbackslash wrap@edcrossref}: \texttt{\textbackslash wrap@edcrossref} is now robust ......................... 192

v1.12.8.
\texttt{\textbackslash flag@end}: \texttt{\textbackslash flag@start} don’t send a error message when a \texttt{\textbackslash edtext} is done without insert (note) but have an endnote .................. 87

v1.13.0.
General: Add \texttt{\textbackslash noteswidthliketwocolumns} and \texttt{\textbackslash notesXwidthliketwocolumns} 32
Added widthliketwocolumns option ........................................ 55
\texttt{\textbackslash newhook@toggle@series}: Add \texttt{\textbackslash newhook@command@toggle@reload} .......... 176
\texttt{\textbackslash para@footsetupX}: In \texttt{\textbackslash para@footsetupX}, use \texttt{\textbackslash columnwidth} instead of \texttt{\textbackslash hsize} 154
\texttt{\textbackslash settoggle@series}: \texttt{\textbackslash settoggle@series} can take an optional arguments to reload series setup. ............................... 174

v1.13.1.
General: Coming back of page and line breaking penalties’s management, deleted by error in v0.17. .......................... 1
Debug quotation environment inside of a \texttt{\textbackslash pstart} preceded by a sectioning command. ......................... 1
\texttt{\textbackslash thepstart}: Add \texttt{\textbackslash @d zeropenalties} in \texttt{\textbackslash pstart} ......................... 103
Change History

v1.13.2.
General: Fix bug with normal footnotes, added by v1.13.0. .......................... 1
\ifledRcol@: Add \if@dpaging for eledpar ................................. 62
v1.13.3.
General: Fix extra spaces with paragraphed footnotes, added by v1.13.0. ........... 1
v1.13.4.
General: Fix bug with index when memoir class is used without hyperref .......... 1
v1.14.0.
General: Debug spurious characters before endnotes. .................................. 161
Delete previous override of \@d@@or\indexh@p at the beginning of a document when hyperref is not loaded. ................................. 214
Moves gobbling command ............................................................ 61
Provide \@gobblefour ................................................................. 61
\edindex: Let eledmac take advantage of imakeidx even when memoir class is used ................................................................. 209
\@ssect: Debug sectioning commands when using both handout and hyperref package. ................................................................. 249
v1.14.2.
\@ssect: Debug \edtext after starred sectioning commands when using memoir class. ................................................................. 246
v1.15.0.
\@edtext@level: New boolean \if@edtext@ ........................................... 93
General: Fix bug with footnotes layout when using some options of the geometry package (bug add by v1.13.0). ................................................................. 1
New commands \AtEveryPstart and \AtEveryPend. ................................ 14
New tools to prevent ambiguous references in lemma ................................... 22
@endsub: Restore subline feature (disabled by mistake in v1.8.0). ................. 87
\footparagraphX: Correct bug with paragraphed familiar footnotes setting. .... 154
\if@lemmacommand@: New boolean \iflemmacommand@ .................................. 96
v1.15.1.
\line@list@stuff: Revert modification of 1.5.2 which makes bug with numbering. Leave vertical mode to solve spurious space before minipage. ................. 86
v1.16.0.
General: Compatibility of standard footnotes with some biblatex styles. ....... 1
New \spanzaindent command ......................................................... 1
\critext: \critext and \edtext are now defined only in eledmac, not in eledpar.
Debug wrong numbering when using \sameword + eledpar + \tag command. ..... 93
v1.16.1.
\lineref: \lineref is not defined if defined by some other package, like lineno.
Eledmac provides \edlineref instead ............................................. 193
v1.17.0.
\critext: The historical \critext now just refers to \edtext (code refactoring). 93
\edtext: Error message when calling \edtext outside of a numbered paragraph. 93
v1.18.0.
\edindex@hyperref: Fix spurious space with \edindex when using imakeidx/indextools + hyperref. ................................................................. 213
General: Add \pstartinfootnoteeverytime ..................................... 26
Compatibility with LuaLaTeX RTL languages ......................................... 1
Change History

Debug `\onlypstartinfootnote` when using `\numberonlyfirstinline` and the current line number differs from the previous. .......................... 26
`\edlabel`: `\edlabel` is now defined only one time for both eledmac and eledpar 189
`\ifledRcol@`: Add `\if1@\dprintingpages` and `\@\dprintingcolumns` for eledpar 62
`\L@\d@section`: Option parapparatus works for endnotes. .................. 162
`\print@line`: Compatibility with LuaL A \TEX R TL languages. ......... 108
`\printlinefootnote`: Code refactoring in `\printlinefootnote`: the printing of the numbers are factorized in `\printlinefootnotearea` ............ 180
`\printpstart`: Debug `\pstartinfootnote` with parallel pages and columns (eledpar) .............................................................. 125

v1.19.0.

General: `\maxhXnotes` and `\maxhnotesX` work now for both two-columns and three-columns setting. .......................... 1
Compatibility with eledpar v.1.13.0. .................................. 1
`\footsplitskips`: `\footsplitskips` doesn’t set `\floatingpenalty` to `\@MM` when processing parallel pages. .................. 123
`\xxref`: `\xxref` works also with right side numbers, when `\Rlineflag` is not empty. .......................... 194

v1.19.1.

General: Call `\correct@footinsX@box` and `\correct@Xfootins@box` directly in `\print@notesX@forpages` and `\print@Xnotes@forpages`, that is in eledpar. 1

v1.20.0.

General: Add `\boxXendlinenum` .............................................. 28
Add `\twolines` and `\morethantwolines` hooks .......................... 25
Add series option. .............................................................. 1
Correct `\inplaceofnumber` hook. ....................................... 1
Explicit error message when calling `\Xfootnote` outside of `\edtext`. 1
Fix bug with line number typesetting direction when using `\eledsection` and similar commands for RTL texts with LuaLaTEX. .............. 1
Fix issues with RTL text in notes when using LuaLaTEX. .............. 1
Options fulllines in `\Xfootnote`. ....................................... 19
The `\newifs` are not followed by boolean values set to false, because it is the \TeX default setting. .......................... 1
`\printlines`: Added `\if0@\d@morethantwolines` and `\if0@\d@morethantwolines` to `\printlines` .......................... 129
`\stanza`: `&` and `\&` can be preceded by spaces. .................. 220
`\xxref`: Debug `\xxref` when not loading eledpar (fix bug added in 1.19.0). 194

v1.21.0.

`\@edindex@hyperref`: Look at the hyperindex option of hyperref before inserting hyperref .............................................................. 213

General: `\AtEveryPstart` and `\AtEveryPend` are now compatible with `\autopar` 1
`\afterXrule` and `\afterruleX` features no longer create problems of overflowing at the bottom of the page. .......................... 1
`\chapter` inside optional argument of `\pstart` works when typesetting parallel pages .............................................................. 1
`\preXnotes` and `\prenotesX` features no longer create problems of overflowing at the bottom of the page. .......................... 1
`\seriesatbegin` and `\seriesatbegin` more efficient .......................... 173
Add `\applabel` and related ................................................. 40
Add \beforenotesX and \beforeXnotes features for notes set in two and three column. 1
Add \hideningering 18
Add \twolinesbutnotmore and \twolinesonlyinsamepage. 1
Add \Xcolalign and \colalignX hooks 30
Add \Xendtwolines, \Xendmorethantwolines, \Xendtwolinesbutnotmore and \Xendtwolinesonlyinsamepage. 26
Add \Xparindent and \hangindentX 29
Add nocritical, noend, nofamiliar and noledgroup options. 1
Add noeledsec package option 1
Debug \beforenotesX maxhnotesX \maxhnotesX \notesXwidthliketwocolumns and \afterruleX with footnotes set in two and three columns. 1
Fix bug when a \Xfootnote follows a \Xendnote in the second argument of \edtext (bug added in eledmac 1.0.0 ). 1
Fix bug with \maxhnotesX when using \foottwocolX or \foottwocolX 1
Fix bug with space between columns with notes in two columns (bug added in v1.13.0). 1
Fix spurious space after first page number in \doendnotes. oldprintnpnumsplace option allows to come back to previous setting. 1
parapparatus option works now with familiar footnotes. 1
Provide \@gobblefive 61
\l@d@section: \endnotes take five arguments. 162
\ledinnotemark: Add \ledinnotemark. 208
\\@num: \@num@ref deleted 82
\\@num defined only one time for both eledmac and eledpar 82
\newhookcommand@series: \newhookcommand@series can take an optional argument. 175
\newhooktoggle@series: \newhooktoggle@series can take an optional argument. 175
\noendnotes: \noendnotes deprecated, prefer noend option. 166
\normalfootfmt: \ledsetnormalparstuff is deprecated and becomes \ledsetnormalparstuffX and \Xledsetnormalparstuff. 124
\print@footnoteXrule: Code refactoring: the spaces after the footnote rules are directly managed in \print@Xfootnoterule and \print@footnoteXrule . 158
\seriesatend: Fix spurious space in \seriesatend 173
\skipnumbering: \skipnumbering defined only one time for both eledmac and eledpar. 89
Correct \skipnumbering for stanza. 89
Delete \skipnumbering@reg. 89
v1.22.0.
General: Add \doendnotesbysection command. 20
Add option for lemma separator inside endnotes 28
Adds hyperlink for references to notes in indices. 1
Fix conflict between noend package option and edtabularx environments 1
Provides support for xindy. 1
Standardize endnotes handbook. 20
When using hyperref package, internal links in index or with \edlineref are now targeted to the top and not longer to the bottom of the lines they refer to. 1
\ledinnote: \ledinnote takes a first optional argument, which is the label for hyperlinks. .................................................. 208

v1.22.1.
General: Fix bug (added on v1.22.0) with \inplaceofnumber hook. ... 1
\prevpage\num: Correct double symbol when using both \parafootsep and \symlinenum. ..................................................... 139
\textbardbl: Robustify \textbardbl ....................................... 179

v1.23.0.
\edtext@level: The boolean \if@edtext@ becomes the counter \edtext@level.
................................................................. 93
General: Add \boxlinenumalign and \boxXendlinenumalign. .... 27
Add \boxstartlinenum, \boxXendstartlinenum, \boxendlinenum, \boxXendendlinenum.
................................................................. 27
Allow use of \sameword with inputenc managing of UTF-8. ...... 1
Compatibility between nofamiliar/nocriticals option and minipage/ledgroup. ... 1
Error message when using \beginnumbering..\endnumbering without \pstart.
................................................................. 1
Fix bug with \sameword when the lemma overlaps multiple line. ... 22
Fix bug with \sameword when the same lemma is used for multiple notes or for nested \edtext. ................................................. 22
Fix bug with \skipnumbering called immediately after a \pstart. ... 1
Fix error of \iftrue not closed. .................................... 1
Fix spurious space with \skipnumbering (bug added on v1.21.0). ... 1
New tools to ensure the line-list file uses the right version of commands when upgrading the eledmac version. ......................................... 1
Optional argument of \sameword can be a comma separated list of \edtext depth. .................................................. 22
\lemma: Fix spurious space after \lemma command ............... 96
\newseries@: Prevent spurious spaces when \Afootnote and similar commands are followed by spaces (bug added on 1.0.0). ......................... 169
\sameword: In order to allow use of \sameword with inputenc, we detokenize its mandatory argument before using it in control sequence names. ........ 100

v1.23.1.
General: Fix bug with \lemma command in the right side. ....... 1

v1.23.2.
General: Compatibility with \LaTeX's release 2015. ............... 1