

# reledmac

## Typeset scholarly editions with L<sup>A</sup>T<sub>E</sub>X\*

Maïeul Rouquette<sup>†</sup>

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

The reledmac provides many tool in order to typeset scholarly edition. It is base on the eledmac package, which was based on ledmac package, which was based on edmac T<sub>E</sub>X package.

It can be use in combination with reledpar in order to typeset two texts in parallel, like an original text and its translation in modern language.

reledmac provides many tools and options. Normally, they are all documented 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 reledmac mail list in:

<http://geekographie.maieul.net/146>

## Contents

<b>1 Introduction</b>	<b>9</b>
1.1 Aim of the package	9
1.2 History	10
1.2.1 edmac	10
1.2.2 ledmac	12
1.2.3 eledmac	12

---

\*This file (reledmac.dtx) has version number v2.0.0, last revised 2015/07/19.

<sup>†</sup>maieul at maieul dot net

1.2.4 reledmac . . . . .	12
1.3 List of works edited with (r)(e)ledmac . . . . .	12
<b>2 How the package works</b>	<b>12</b>
<b>3 Options</b>	<b>13</b>
3.1 Specific features . . . . .	13
3.2 Optimize package's performance . . . . .	13
<b>4 Text lines and paragraphs numbering</b>	<b>14</b>
4.1 Text lines numbering . . . . .	14
4.2 Paragraphs . . . . .	15
4.2.1 Basis . . . . .	15
4.2.2 Producing automatically \pstart...\pend . . . . .	15
4.2.3 Content before specific \pstart and after specific \pend . . . . .	16
4.2.4 Content before every \pstart and after every \pend . . . . .	16
4.2.5 Numbering paragraphs (\pstart) . . . . .	16
4.2.6 Languages written in Right to Left . . . . .	17
4.2.7 Memory limits . . . . .	17
4.3 Lineation commands . . . . .	17
4.3.1 Disabling lineation . . . . .	17
4.3.2 Setting lineation start and step . . . . .	18
4.3.3 Setting lineation reset . . . . .	18
4.3.4 Setting line number margin . . . . .	18
4.3.5 Other settings . . . . .	19
4.4 Changing the line numbers . . . . .	19
4.4.1 Sublineation . . . . .	19
4.4.2 Locking lineation . . . . .	19
4.4.3 Setting and changing line number . . . . .	19
4.4.4 Line number style . . . . .	20
4.4.5 Skipping and hiding number . . . . .	20
4.4.6 Execute code at each line . . . . .	20
<b>5 Apparatus commands</b>	<b>21</b>
5.1 Terminology . . . . .	21
5.2 Critical notes . . . . .	21
5.2.1 The lemma . . . . .	21
5.2.2 Footnotes . . . . .	22
5.2.3 Endnotes . . . . .	22
5.2.4 Paragraph in critical apparatus . . . . .	23
5.2.5 Change lemma and line number . . . . .	23
5.2.6 Changing the names of commands for critical apparatus . . . . .	24
5.3 Disambiguation of identical words in the apparatus . . . . .	24
5.3.1 Basic use . . . . .	25
5.3.2 Notes about input encoding with UTF-8 processor . . . . .	25
5.3.3 Use with \lemma command . . . . .	25

5.3.4 Customizing . . . . .	27
5.4 Familiar notes . . . . .	27
5.4.1 Basic use . . . . .	27
5.4.2 Customizing mark . . . . .	27
5.4.3 Separator for multiple footnotes . . . . .	27
5.5 Changing series . . . . .	28
5.5.1 Create a new series . . . . .	28
5.5.2 Delete series . . . . .	28
5.5.3 Series order . . . . .	28
5.6 Position of critical and familiar footnotes . . . . .	28
<b>6 Critical apparatus appearance</b> . . . . .	<b>28</b>
6.1 Notes arrangement in a series . . . . .	29
6.2 Control line number printing . . . . .	29
6.2.1 Print line number only at first time . . . . .	29
6.2.2 Abbreviate line range . . . . .	30
6.2.3 Disable line number . . . . .	30
6.2.4 Printing pstart number . . . . .	31
6.2.5 Space around number . . . . .	31
6.2.6 Space around line symbol . . . . .	31
6.2.7 Space in place of number . . . . .	31
6.2.8 Boxing line number and line symbol . . . . .	32
6.3 Separator between the lemma and the note . . . . .	32
6.3.1 For footnotes . . . . .	32
6.3.2 For endnotes . . . . .	33
6.4 Font style . . . . .	33
6.4.1 For line number . . . . .	33
6.4.2 For the lemma . . . . .	34
6.5 Styles of notes content . . . . .	34
6.6 Arbitrary code at the beginning of notes . . . . .	34
6.7 Options for footnotes in columns . . . . .	35
6.7.1 Alignment . . . . .	35
6.7.2 Size of the columns . . . . .	35
6.8 Options for paragraphed footnotes . . . . .	35
6.8.1 Mark separation of notes . . . . .	35
6.8.2 Ragging . . . . .	36
6.9 Options for block of notes . . . . .	36
6.9.1 Text before notes . . . . .	36
6.9.2 Spacing . . . . .	36
6.9.3 Rule . . . . .	36
6.9.4 Maximum height . . . . .	37
6.10 Footnotes and the reledpar columns . . . . .	37
6.11 Endnotes in one paragraph . . . . .	37
<b>7 Fonts</b> . . . . .	<b>37</b>

<b>8 Verse</b>	<b>38</b>
8.1 Basic . . . . .	38
8.2 Define stanza indents . . . . .	38
8.3 Repeating stanza indents . . . . .	39
8.4 Manual stanza indent . . . . .	40
8.5 Stanza breaking . . . . .	40
8.6 Hanging symbol . . . . .	40
8.7 Long verse and page break . . . . .	41
8.8 Hanging symbol . . . . .	41
8.9 Content before/after verses . . . . .	41
8.10 Various tools . . . . .	41
<b>9 Grouping</b>	<b>42</b>
<b>10 Cross referencing</b>	<b>42</b>
10.1 Basic use . . . . .	42
10.2 Refer to a critical notes . . . . .	43
10.3 Cross-referencing which return a number in any case . . . . .	43
10.3.1 Cross-referencing in order to define line number of a critical note . . . . .	44
10.4 Not automatic cross-referencing . . . . .	44
10.5 Normal $\LaTeX$ cross-referencing . . . . .	44
10.6 References to lines commented in the apparatus . . . . .	44
<b>11 Side notes</b>	<b>45</b>
11.1 Basis . . . . .	45
11.2 Setting . . . . .	46
11.2.1 Width . . . . .	46
11.2.2 Vertical position . . . . .	46
11.2.3 Distance to the main text . . . . .	46
11.2.4 Separator between notes . . . . .	46
<b>12 Indexing</b>	<b>46</b>
12.1 Basis . . . . .	46
12.2 Separator between page and line numbers . . . . .	47
12.3 Using xindy . . . . .	47
12.4 Advanced setting . . . . .	48
<b>13 Tabular material</b>	<b>48</b>
<b>14 Sectioning commands</b>	<b>51</b>
14.1 Sectioning commands without line numbers or critical notes . . . . .	51
14.2 Sectioning commands with line numbering and critical notes . . . . .	52
14.3 Optimization . . . . .	52
<b>15 Quotation environments</b>	<b>53</b>

<b>16 Page breaks</b>	<b>53</b>
16.1 Control page breaking . . . . .	53
16.2 Prevent page break in a long verses . . . . .	53
<b>17 Miscellaneous</b>	<b>54</b>
17.1 Known and suspected limitations . . . . .	54
17.2 ‘No room for a new’ . . . . .	54
17.2.1 Marginal notes . . . . .	55
17.2.2 Paragraph shape . . . . .	55
17.2.3 Paragraphed footnotes . . . . .	55
17.2.4 Use with other packages . . . . .	55
17.3 Parallel typesetting . . . . .	56
<b>I Implementation overview</b>	<b>57</b>
<b>II Preliminaries</b>	<b>57</b>
II.1 Links with original edmac . . . . .	57
II.2 Package declaration . . . . .	57
II.3 Package options . . . . .	58
II.4 Loading packages . . . . .	59
II.5 Boolean flags . . . . .	59
II.6 Messages . . . . .	60
II.7 Gobbling . . . . .	64
II.8 Miscellaneous commands . . . . .	64
II.9 Prepare reledpar . . . . .	65
<b>III Sectioning commands</b>	<b>65</b>
<b>IV List macros</b>	<b>69</b>
<b>V Line counting</b>	<b>70</b>
V.1 Choosing the system of lineation . . . . .	70
V.2 Line number margin . . . . .	71
V.3 Line number initialization and increment . . . . .	72
V.4 Line number locking . . . . .	73
V.5 Line number style . . . . .	74
V.6 Line number printing . . . . .	75
V.7 Line number counters and lists . . . . .	76
V.8 Line number locking counter . . . . .	77
V.9 Line number associated to lemma . . . . .	77
V.10 Reading the line-list file . . . . .	80
V.11 Commands within the line-list file . . . . .	82
V.12 Writing to the line-list file . . . . .	92

<b>VI Marking text for notes</b>	<b>96</b>
VI.1 <code>\edtext</code> itself . . . . .	97
VI.2 Substitute lemma . . . . .	103
VI.3 Substitute line numbers . . . . .	103
VI.4 Lemma disambiguation . . . . .	104
<b>VII Paragraph decomposition and reassembly</b>	<b>109</b>
VII.1 Boxes, counters, <code>\pstart</code> and <code>\pend</code> . . . . .	109
VII.2 Processing one line . . . . .	114
VII.2.1 General process . . . . .	114
VII.2.2 Process for “normal” line . . . . .	115
VII.2.3 Process for line containing <code>\eledsection</code> command . . . . .	116
VII.2.4 Hooks . . . . .	116
VII.2.5 Sidenotes and marginal line number initialization . . . . .	116
<b>VIII Line and page number computation</b>	<b>117</b>
<b>IX Line number printing</b>	<b>120</b>
<b>X Pstart number printing in side</b>	<b>123</b>
<b>XI Restoring footnotes and penalties</b>	<b>124</b>
XI.1 Add insertions to the vertical list . . . . .	125
XI.2 Penalties . . . . .	126
XI.3 Printing leftover notes . . . . .	126
<b>XII Critical footnotes</b>	<b>127</b>
XII.1 Fonts . . . . .	127
XII.2 Individual note options . . . . .	128
XII.3 Notes language . . . . .	128
XII.4 General survey of the way we manage notes . . . . .	129
XII.5 General setup . . . . .	129
XII.6 Footnotes arrangement . . . . .	130
XII.6.1 User level macro . . . . .	130
XII.6.2 Normal footnote . . . . .	130
XII.6.3 Paragraphed footnotes . . . . .	134
XII.6.4 Columnar footnotes . . . . .	140
XII.7 Critical notes presentation . . . . .	145
XII.7.1 Font tools . . . . .	145
XII.7.2 Pstart number in footnote . . . . .	146
XII.7.3 Line number printing . . . . .	146
<b>XIII Familiar footnotes</b>	<b>153</b>
XIII.1 Adjacent footnotes . . . . .	153
XIII.2 Regular footnotes for numbered texts . . . . .	154
XIII.3 Footnote formats . . . . .	155
XIII.4 Footnote arrangement . . . . .	155

XIII.4.1 User level macro . . . . .	155
XIII.4.2 Normal footnotes . . . . .	156
XIII.4.3 Two columns footnotes . . . . .	160
XIII.4.4 Three columns footnotes . . . . .	162
XIII.4.5 Paragraphed footnotes . . . . .	163
<b>XIV Code common to both critical and familiar footnote in normal arrangement</b>	<b>167</b>
<b>XV Footnotes' width for two columns</b>	<b>167</b>
<b>XVI Footnotes' order</b>	<b>169</b>
<b>XVII Footnotes' rule</b>	<b>169</b>
<b>XVIII Specific skip for first series of footnotes</b>	<b>169</b>
XVIII.0.1 Overview . . . . .	169
XVIII.0.2 User level command . . . . .	170
XVIII.0.3 Internal commands . . . . .	170
<b>XIX Endnotes</b>	<b>171</b>
<b>XX Generate series of notes</b>	<b>177</b>
XX.1 Test if series is still existing . . . . .	177
XX.2 Init specific to reledpar . . . . .	177
XX.3 For critical footnotes . . . . .	178
XX.3.1 Options . . . . .	178
XX.3.2 Create inserts, needed to add notes in foot . . . . .	179
XX.3.3 Create commands for critical apparatus, \Afootnote, \Bfootnote etc. . . . .	179
XX.3.4 Set standard display . . . . .	180
XX.4 For familiar footnotes . . . . .	180
XX.4.1 Options . . . . .	180
XX.4.2 Create tools for familiar footnotes (\footnoteX) . . . . .	181
XX.5 The endnotes . . . . .	181
XX.5.1 The auxiliary file . . . . .	182
XX.5.2 The main macro . . . . .	182
XX.5.3 The options . . . . .	182
XX.6 Init standards series (A,B,C,D,E) . . . . .	183
<b>XXI Setting series display</b>	<b>183</b>
XXI.1 Change series order . . . . .	183
XXI.2 Test series order . . . . .	184
XXI.2.1 Get the first series . . . . .	184
XXI.3 Series setting . . . . .	184
XXI.3.1 General way of working . . . . .	184
XXI.3.2 Tools to set options . . . . .	185

XXI.3.3 Tools to generate options commands . . . . .	186
XXI.3.4 Options for critical notes . . . . .	187
XXI.3.5 Options for familiar notes . . . . .	188
XXI.3.6 Options for endnotes . . . . .	189
XXI.4 Hooks for a particular footnote . . . . .	189
XXI.5 Alias . . . . .	190
<b>XXII Output routine</b>	<b>190</b>
XXII.0.1 Page number management . . . . .	190
XXII.0.2 Extra footnotes output . . . . .	190
XXII.0.3 Standard output's commands patching . . . . .	193
<b>XXIII Cross referencing</b>	<b>195</b>
<b>XXIV Side notes</b>	<b>202</b>
<b>XXV Minipages and such</b>	<b>208</b>
<b>XXVI Indexing</b>	<b>212</b>
<b>XXVII Verse</b>	<b>218</b>
XXVII.1 Hanging symbol management . . . . .	218
XXVII.2 Using & character . . . . .	219
XXVII.3 Code category setting . . . . .	219
XXVII.4 Stanza count and indent . . . . .	219
XXVII.5 Main work . . . . .	221
XXVII.6 Restore catcode and penalties . . . . .	222
<b>XXVIII Arrays and tables</b>	<b>222</b>
XXVIII.1 Preamble: macro as environment . . . . .	222
XXVIII.2 Tabular environments . . . . .	226
XXVIII.2.1 Disabling and restoring commands . . . . .	226
XXVIII.2.2 Counters, boxes and lengths . . . . .	229
XXVIII.2.3 Tabular typesetting . . . . .	232
XXVIII.2.4 Environments . . . . .	242
<b>XXIX Quotation's commands</b>	<b>242</b>
<b>XXX Section's title commands</b>	<b>243</b>
XXX.1 Commands to disable some feature . . . . .	243
XXX.2 General overview . . . . .	243
XXX.3 \beforeeledchapter command . . . . .	244
XXX.4 Auxiliary commands . . . . .	244
XXX.5 Patching standard commands . . . . .	245
XXX.6 Main code of \eledxxx commands . . . . .	249
XXX.7 Macros written in the auxiliary file . . . . .	252



<b>XXXI Page breaking or no page breaking depending of specific lines</b>	<b>253</b>
<b>XXXII Long verse: prevents being separated by a page break</b>	<b>255</b>
<b>XXXIII Compatibility with eledmac</b>	<b>255</b>
<b>Appendix A Some things to do when changing version</b>	<b>258</b>
Appendix A.1 Migrating from edmac to ledmac . . . . .	258
Appendix A.2 Migration from ledmac to eledmac . . . . .	259
Appendix A.3 Migration to eledmac 1.5.1 . . . . .	260
Appendix A.4 Migration to eledmac 1.12.0 . . . . .	260
Appendix A.5 Migration to eledmac 17.1 . . . . .	261
Appendix A.6 Migration to eledmac 1.21.0 . . . . .	261
Appendix A.6.1 \Xledsetnormalparstuff and \ledsetnormalparstuffX	261
Appendix A.6.2 Endnotes . . . . .	261
Appendix A.7 Migration to eledmac 1.22.0 . . . . .	261
Appendix A.8 Migration to eledmac 1.23.0 . . . . .	261
Appendix A.9 Migration from eledmac to reledmac . . . . .	262
Appendix A.9.1 Risk of ‘no room for a new’ . . . . .	262
Appendix A.9.2 Multiple indices with memoir . . . . .	262
Appendix A.9.3 Deprecated commands and options . . . . .	262
Appendix A.9.4 \renewcommand replaced by command . . . . .	263
Appendix A.9.5 Commands the names of which have been changed . . .	263
Appendix A.9.6 Endnotes . . . . .	265
Appendix A.9.7 Z Series . . . . .	265
Appendix A.9.8 Internal commands . . . . .	265
<b>References</b>	<b>266</b>
<b>Index</b>	<b>266</b>
<b>Change History</b>	<b>301</b>

## 1 Introduction

### 1.1 Aim of the package

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, section or paragraph;
- 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.  $\text{\LaTeX}$  and Eledmac will take care of the formatting and visual correlation of all the disparate types of information.

Apart from reledmac there are some other  $\text{\LaTeX}$  packages for critical edition typesetting. However, the aim of reledmac is to provide a “all in one” and flexible tool in the field of critical edition.

Any suggestion for new features are welcome.

This manual contains a general description of how to use reledmac and the complete source code for the package, with extensive documentation (in sections I and following, numbered in Roman numeric) It finish by a list of action to do when migrating from some version to other, a historical of the change and an index to the source code.

We do not suggest that you need to read the source code for this package in order to use it; we provide this code primarily for reference, and many of our comments on it repeat material that is also found in the earlier sections.

But no documentation, however thorough, can cover every question that comes up, and many can be answered quickly by consultation of the code. On a first reading, we suggest that you should read only the general documentation in sections 2, unless you are particularly interested in the innards of reledmac.

## 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.<sup>1</sup> A description by John and Dominik of this version of edmac was published as ‘An overview of edmac: a PLAIN T<sub>E</sub>X 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 T<sub>E</sub>X 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.

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*,<sup>2</sup> an edition of the letters of Nicolaus Copernicus,<sup>3</sup> Simon Bredon’s *Arithmetica*,<sup>4</sup> a Latin translation by Plato of Tivoli of an Arabic astrolabe text,<sup>5</sup> a Latin translation of part II of the Arabic *Algebra* by Abū Kāmil Shujā’ b. Aslam,<sup>6</sup> the Latin *Rithmachia* of Werinher von Tegernsee,<sup>7</sup> a middle-Dutch romance epic on the Crusades,<sup>8</sup> a seventeenth-century Hungarian politico-philosophical tract,<sup>9</sup> an anonymous Latin compilation from Hungary entitled *Sermones Compilati in Studio Generali Quinqueecclesiensi in Regno Ungarie*,<sup>10</sup> the collected letters and papers of Leibniz,<sup>11</sup> Theodosius’s *Spherics*, the German *Algorismus* of Sacrobosco, the Sanskrit text of the *Kāśikāvṛtti* of Vāmana and Jayāditya,<sup>12</sup> and the English texts of Thomas Middleton’s collected works.

<sup>1</sup>This version of the macros was used to format the Sanskrit text in volume I of *Metarules of Pāṇinian Grammar* by Dominik Wujastyk (Groningen: Forsten, 1993).

<sup>2</sup>Gerhard Brey used edmac in the production of Hubert L. L. Busard and Menso Folkerts, *Robert of Chester’s (?) Redaction of Euclid’s Elements, the so-called Adelard II Version*, 2 vols., (Basel, Boston, Berlin: Birkhäuser, 1992).

<sup>3</sup>Being prepared at the German Copernicus Research Institute, Munich.

<sup>4</sup>Being prepared by Menso Folkerts *et al.*, at the Institut für Geschichte der Naturwissenschaften in Munich.

<sup>5</sup>Richard Lorch, Gerhard Brey *et al.*, at the same Institute.

<sup>6</sup>Richard Lorch, ‘Abū Kāmil on the Pentagon and Decagon’ in *Vestigia Mathematica*, ed. M. Folkerts and J. P. Hogendijk (Amsterdam, Atlanta: Rodopi, 1993).

<sup>7</sup>Menso Folkerts, ‘Die *Rithmachia* des Werinher von Tegernsee’, *ibid.*

<sup>8</sup>Geert H. M. Claassens, *De Middelnederlandse Kruisvaartromans*, (Amsterdam: Schiphower en Brinkman, 1993).

<sup>9</sup>Emil Hargittay, *Csáky István: Politica philosophiai Okoskodás-szerint való rendes életnek példája (1664–1674)* (Budapest: Argumentum Kiadó, 1992).

<sup>10</sup>Being produced, as was the previous book, by Gyula Mayer in Budapest.

<sup>11</sup>Leibniz, *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>)

<sup>12</sup>Being prepared at Poona and Lausanne Universities.

### 1.2.2 ledmac

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 ( $\LaTeX$  edmac).

Since July 2011, ledmac is maintained by Maïeul Rouquette. Is it more and more powerful and flexible, but also more and more divergent from original TeX macro.

### 1.2.3 eledmac

Important changes were put in version 1.0, to make ledmac more easily extensible (see 6 p. 28). These changes can trigger small problems with the old customization.

That is why a new name was selected: eledmac (extended ledmac).

To migrate from ledmac to eledmac, please read Appendix Appendix A.2 (p. 259).

### 1.2.4 reledmac

eledmac has facilitated the creation of customized critical edition. However, this was made in a non systematic way in method to customize them.

In other side, many deprecated commands were kept, and many technical debt was accumulated, blocking the futur evolution.

For all these reasons, Maïeul Rouquette decided to make a spring cleaning. As some commands name were changed, the ascendant compatibility was (a little) broken.

A new name was selected: reledmac (extended renewed eledmac).

To migrate from eledmac to reledmac, please read Appendix ?? (p. ??).

## 1.3 List of works edited with (r)(e)ledmac

A collaborative list of works edited with (r)(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 (r)(e)ledmac.

## 2 How the package works

The reledmac package is a three-pass package like  $\LaTeX$  itself. Although your textual apparatus and line numbers will be printed even on the first run, it takes two more passes through  $\LaTeX$  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. `reledmac` 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 misnumbered at the top of a page, try running `ℒTEX` 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 will want to print the text that you are editing.

Unnumbered text is not printed with line numbers, and you can't use `reledmac`'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. There are two types of options: options which provide specific features and option which optimize package's performance. It is advised to read the relevant parts of the handbook before reading the first type of option, but you can look at the second type in you first reading of handbook.

### 3.1 Specific features

**`draft`** underlines lemmas in the main text.

**`eledmac-compat`** help to migrate from `eledmac` to `reledmac` (see Appendix A.9.5 p. 263).

**`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 15 p. 53).

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

**`xindy`** and `xindy+hyperref` are for selecting `xindy` as the index processor (12.3 p. 47).

**`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 `reledpar`. This is useful when alternating between normal and parallel typesetting.

### 3.2 Optimize package's performance

**`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. 52).

**noend** disables tools for endnotes (`\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** `reledmac` allows to use of series of critical notes and new series of normal notes inside `minipage` and `ledgroup` environments (see 9 p. 42). 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 `reledmac` faster.

**series** `reledmac` defines five levels of notes: A, B, C, D, E. Using all these levels consumes memory space and processing speed. This is why, if your work does not require all of the A-E 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.

## 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 this:

```
\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>.<series>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 `\beginnumbering` and `\endnumbering` 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.

`reledmac` has to read and store in memory a certain amount of information about the entire section when it encounters a `\beginnumbering` 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

`\pstart` Within a numbered section, each paragraph of numbered text must be marked using the  
`\pend` `\pstart` and `\pend` commands like this:

```
\pstart
Paragraph of text.
\pend
```

Text that appears within a numbered section but is not marked with `\pstart` and `\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:

```
\beginnumbering
\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
\endnumbering
```

### 4.2.2 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.<sup>13</sup>

#### 4.2.3 Content before specific `\pstart` and after specific `\pend`

`\AtEveryPstart` Both `\pstart` and `\pend` can take an optional argument, in brackets. Its content will be printed before the beginning of `\pstart` / after the end of `\pend` instead of the argument of `\AtEveryPstart` / `\AtEveryPend`. If you need to start a `\pstart` by brackets, or to add brackets after a `\pend`, just add a `\relax` between `\pstart`/`\pend` and the brackets.

. This feature is also useful when typesetting verses (see 8 p. 38) or `eledpar` (see 17.3 p. 56).

A `\noindent` is automatically added before this argument.

#### 4.2.4 Content before every `\pstart` and after every `\pend`

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

`\AtEveryPend`

#### 4.2.5 Numbering paragraphs (`\pstart`)

`\numberpstarttrue` 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`.  
`\numberpstartfalse` You can redefine the command `\thepstart` to change style. You can change the value of the `pstart` number by using *after* `\beginnumbering`:  
`\thepstart` `\setcounter{numberpstart}{value}`

On each `\beginnumbering` the numbering restarts.

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

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

<sup>13</sup>For a detailed study of the reasons for this restriction, see Barbara Beeton, 'Initiation rites', *TUGboat* 12 (1991), pp. 257–258.



### 4.2.6 Languages written in Right to Left

If you use languages written in right to left, we Lua $\TeX$  or Xe $\TeX$ , so you must switch text direction `\before` the `\pstart` command.

### 4.2.7 Memory limits

`\pausenumbering`  
`\resumenumbering`

**This paragraph is kept for history, but problem described below should not appear with recent version of  $\TeX$ .** `reledmac` 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  $\TeX$  may reach its memory limit. There are two solutions to this. The first is to get a larger  $\TeX$  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 `\memorybreak` between the relevant `\pend` and `\pstart`.

## 4.3 Lineation commands

### 4.3.1 Disabling lineation

`\numberlinefalse` Line numbering can be disabled with `\numberlinefalse`. It can be enabled again with  
`\numberlinetrue` `\numberlinetrue`.

### 4.3.2 Setting lineation start and step

`\firstlinenum` By default, `reledmac` numbers every 5th line. There are two counters, `firstlinenum` and `linenumincrement`, that control this behaviour; they can be changed using `\firstlinenum{<num>}` and `\linenumincrement{<num>}`. `\firstlinenum` specifies the first line that will have a printed number, and `\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:

```
\firstlinenum{1} \linenumincrement{2}
```

`\firstsublinenum` There are similar commands, `\firstsublinenum{<num>}` and `\sublinenumincrement{<num>}` for controlling sub-line numbering.

`\sublinenumincrement` You can define `\linenumberlist` to specify a non-uniform distribution of printed line numbers. For example:

```
\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 `\linenumberlist` or following the vacuous definition

```
\def\linenumberlist{}
```

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

### 4.3.3 Setting lineation reset

`\lineation` Lines can be numbered either by page, by `pstart` or by section; you specify this using the `\lineation{<arg>}` macro, where `<arg>` is either `page`, `pstart` or `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 `\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

`\linenummargin` The command `\linenummargin{<location>}` specifies the margin where the line (or `pstart`) numbers will be printed. The permissible value for `<location>` is one out of the list `left`, `right`, `inner`, or `outer`, for example `\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` When a marginal line number is to be printed, there are a lot of ways to display it.  
`\rightlinenum` You can redefine `\leftlinenum` and `\rightlinenum` to change the way marginal line  
`\linenumsep` 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.

### 4.4.1 Sublineation

`\startsub` You insert the `\startsub` and `\endsub` commands in your text to turn sub-lineation  
`\endsub` 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.

### 4.4.2 Locking lineation

`\startlock` The `\startlock` command, used in running text, locks the line number at its current  
`\endlock` 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}`.

### 4.4.3 Setting and changing line number

`\setline` In some cases you may want to modify the line numbers that are automatically calculated:  
`\advanceline` 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. `\advance` takes one argument, an amount that should be added to the current line number; it may be positive or negative.

`\setlinenum` The `\setline` and `\advance` 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 `\pend` and a `\pstart`. It sets the line number to `<num>`. It has no effect if used within a `\pstart... \pend` group.

#### 4.4.4 Line number style

`\linenumberstyle` Line numbers are normally printed as arabic numbers. You can use `\linenumberstyle{<style>}`  
`\sublinenumberstyle` 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.

#### 4.4.5 Skipping and hiding number

`\skipnumbering` 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.

`\hidenumbering` 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.

#### 4.4.6 Execute code at each line

`dolinehook` `reledmac` provides to advanced feature for user. The argument passed to `\dolinehook{<arg>}`  
`doinsidelinehook` will be executed before slicing a new line in the paragraph. The argument passed to `\doinsidelinehook{<arg>}` will be executed before printing a new line. In many case, the second of two is more useful than the first. The file `examples/2-line_numbers_in_header.tex` provides an example of use in order to print the first and last line number of a page in the header.

## 5 Apparatus commands

### 5.1 Terminology

We call “critical notes” notes which refer to both a lemma, that is a part of text and a line number. Critical notes are subdivided in critical footnotes and critical endnotes.

We call “familiar notes” notes which refer to a footnote mark in the main text.

reledmac manages many series of notes of each category. A series of notes is identified by an uppercase letter. When the series letter is at the beginning of a command name, it refers to a critical footnote. When the series letter is at the end of a command name, it refers to a critical endnote.

So :

- `\Afootnote` is a critical footnote of the series A.
- `\Bendnote` is a critical footnote of the series B.
- `\footnoteC` is a critical footnote of the series C.

### 5.2 Critical notes

#### 5.2.1 The lemma

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

```
\edtext{<lemma>}{<commands>}
```

The `<lemma>` argument is the lemma in the main text: `\edtext` both prints this as part of the text, and makes it available to the `<commands>` you specify to generate notes.

For example:

I am happy :		1	I am happy : I saw my friend Smith on
I saw my friend <code>\edtext{Smith}{</code>		2	Tuesday.
<code>\Afootnote{Jones C, D.}}</code>			
on Tuesday.			
			1 Smith] Jones C, D.

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 `\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:

I am happy : <code>\edtext{I saw my friend</code>	1	I am happy : I saw my friend Smith on
<code>\edtext{Smith}{\Afootnote{Jones</code>	2	Tuesday.
<code>C, D.}}</code> on Tuesday.}{		
<code>\Bfootnote{The date was</code>		
July 16, 1954.}		1 Smith] Jones C, D.
}		
		1-2 I saw my friend Smith on Tuesday.] The
		date was July 16, 1954.

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

### 5.2.2 Footnotes

The second argument of the `\edtext` macro,  $\langle commands \rangle$ , may contain a series of subsidiary commands that generate various kinds of notes.

`\Afootnote` Five separate series of the footnotes are maintained; each macro takes one argument like `\Afootnote{ $\langle text \rangle$ }`. 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 E notes at the bottom. `\Bfootnote` These are the main macros that you will use to construct the critical apparatus of your text. `\Cfootnote` `\Dfootnote` `\Efootnote`

If you need more series of critical notes, please look at 5.5.1 p. 28.

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:

- `fulllines` to disable `\Xtwolines` and `\Xmorethantwolines` features for this note (cf. 6.2.2 p. 30).
- `nonum` to disable line numbering for this note.
- `nosep` to disable the lemma separator for this note.

Example: `\Afootnote[nonum]{ $\langle text \rangle$ }`.

### 5.2.3 Endnotes

`\Aendnote` The package also maintains five separate series of endnotes.

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

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

`\Aendnote[ $\langle option \rangle$ ]{ $\langle text \rangle$ }`.

$\langle option \rangle$  can contain a comma separated list of values. Allowed values are:

- `fulllines` to disable `\Xendtwolines` and `\Xendmorethantwolines` features for this particular note (cf. 6.2.2 p. 30).
- `nonum` to disable line number for this particular note.
- `nosep` to disable the lemma separator for this particular note.

Normally, endnotes are not printed: you must use the `\doendnotes{ $\langle s \rangle$ }`, where  $\langle s \rangle$  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  $\langle s \rangle$  series are printed, for all numbered section.

`\doendnotesbysection` 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 (6.3.2 p. 33).

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} }
```

#### 5.2.4 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 :

```
\usepackage[parapparatus]{eledmac}
```

#### 5.2.5 Change lemma and line number

`\lemma` 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:

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

1 I am happy : I saw my friend Smith on  
2 Tuesday.  


---

1 Smith ] Jones C, D.  


---

1-2 I ... Tuesday. ] The date was July 16, 1954.

`\linenum` 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 `reledmac` 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 does not 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 (10 p. 42) 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.

### 5.2.6 Changing the names of commands for critical apparatus

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 does not mean you have to type `\Afootnote` when you would 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 at the start of your file:<sup>14</sup>

```
\newcommandx{\variant}[2][1,usedefault]{\Afootnote[#1]{#2}}
\newcommandx{\explanatory}[2][1,usedefault]{\Bfootnote[#1]{#2}}
\newcommand{\trivial}[1]{\Aendnote{#1}}
\newcommandx{\testimonia}[2][1,usedefault]{\Cfootnote[#1]{#2}}
```

## 5.3 Disambiguation of identical words in the apparatus

Sometimes, the same word occurs twice (or more) in the same line. `reledmac` provides tools to disambiguate references in the critical notes. The lemma will be followed by a reference number if a given word occurs more than once in the same line.

---

<sup>14</sup>We use `\newcommand` and `\newcommandx` instead of classical `\let` command because the `edtabular` environments have to modify the notes definition, and we need to use the newest definition of notes. Read the handbook of `xargs` to know more about `\newcommandx`.



### 5.3.1 Basic use

`\sameword` To use this tool, you have to mark every occurrence of the potentially ambiguous term with the `\sameword` command:

```
Lupus \sameword{aut} canis \edtext{\sameword{aut}}{\Afootnote{et}} felix
```

In this example, `aut` will be followed, in the critical note, by the exponent 2 if it is printed in the same line as the first `aut`, but it will not if it is printed in a different line. The number is printed only after the second run.

### 5.3.2 Notes about input encoding with UTF-8 processor

If you use UTF-8 processor, like  $\text{\XeTeX}$  or  $\text{\LuaTeX}$ , 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, “ $\alpha$ ” has two possible unicode numbers:

- GREEK SMALL LETTER ALPHA (U+03B1) + COMBINING GREEK YPOGEGRAMMENI (U+0345)
- 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 `reledmac`, the `\sameword` 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 `\sameword` will process the text inaccurately, based on the unicode numbers. To prevent this, do the following:

- If you use  $\text{\XeTeX}$ , add this line in your preamble: `\XeTeXinputnormalization 1`.
- If you use  $\text{\LuaTeX}$ , use the `uninormalize` package of Michal Hoftich<sup>15</sup> with the `buffer` option set to true.

With these tools,  $\text{\XeTeX}$  /  $\text{\LuaTeX}$  will dynamicaly normalize unicode input when reading the file. Consequently, you will have no problems with the `\sameword` command.

### 5.3.3 Use with `\lemma` command

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

For example in the following example:

```
some thing
\edtext{\sameword{sw}}
```

---

<sup>15</sup><https://github.com/michal-h21/uninormalize>.

```

        and other \sameword{sw}
        and again \sameword{sw}
        it is all}%
    }{\lemma{\sameword{sw} \ldots all}\Afootnote{critical note}}.%

```

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

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

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

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

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

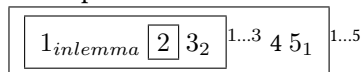
```

\edtext{some \edtext{\sameword[1]{word}}\Afootnote{om. M}}
    and other \sameword{word}
    and again a \sameword{word}
    it is all}%
}{\lemma{some \sameword{word} \ldots all}\Afootnote{critical note}}.%

```

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

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



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

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

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

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

### 5.3.4 Customizing

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

```
\newcommand{\showwordrank}[2]{%
  #1\textsuperscript{#2}%
}
```

## 5.4 Familiar notes

### 5.4.1 Basic use

`\footnoteA` As well as the standard L<sup>A</sup>T<sub>E</sub>X footnotes generated via `\footnote`, the package also provides five series of additional footnotes called `\footnoteA` through `\footnoteE`. These

`\footnoteB` have the familiar marker in the text, and the marked text at the foot of the page can be

`\footnoteC` formatted using any of the styles described for the critical footnotes. Note that the ‘regular’

`\footnoteD` footnotes have the series letter at the end of the macro name whereas the critical

`\footnoteE` footnotes have the series letter at the start of the name.

### 5.4.2 Customizing mark

`\thefootnoteA` Each series uses a set of macros for styling the marks. The mark numbering scheme of

`\bodyfootmarkA` series A is defined by the `\thefootnoteA` macro; the default is:

`\footfootmarkA` `\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.

### 5.4.3 Separator for multiple footnotes

The `footmisc` package [Fai03] by Robin Fairbairns has an option whereby sequential footnote marks in the text can be separated by commas<sup>3,4</sup> like so. As a convenience `Eledmac` provides this automatically.

`\multfootsep` `\multfootsep` is used as the separator between footnote markers. Its default definition is:

```
\providecommand*{\multfootsep}{\textsuperscript{\normalfont,,}}
```

and can be changed if necessary.

## 5.5 Changing series

### 5.5.1 Create a new series

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

### 5.5.2 Delete series

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

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

### 5.5.3 Series order

The default series order is the one called with the `series` option of the package, or, if this option is not used, A, B, C, D, E. 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 `\seriesatbegin{<s>}` to pull up a given series `<s>` to the beginning, or `\seriesatend{<s>}` to push it down to the end.

## 5.6 Position of critical and familiar footnotes

`\fnpos` There is a historical incoherence in (r)(e)ledmac. 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}
```

## 6 Critical apparatus appearance

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. If the optional argument is omitted or empty, the setting will concern all series.

When a length, noted `<l>`, is used, it can be stretchable: `a plus b minus c`. The final length `m` is calculated by  $\TeX$  to have:  $a - c \leq m \leq a + b$ . If you use some relative unity<sup>16</sup>, it will be relative to fontsize of the footnote, except for commands concerning

---

<sup>16</sup>Like `em` which is the width of a `mg`.

the place kept by the notes — including blank space.

There is also name convention:

- Names prefixed by X are for setting of critical footnotes.
- Names prefixed by Xend are for setting of critical endnotes.
- Names suffixed by X are for setting of familiar footnotes.

## 6.1 Notes arrangement in a series

`\Xarrangement` By default, all footnotes are formatted as a series of separate paragraphs in one column.  
`\arrangementX` Three other formats are also available for notes.

Use `\Xarrangement[\langle s \rangle]{\langle a \rangle}` to change the arrangement of the  $\langle s \rangle$  series of critical footnotes and `\arrangementX[\langle s \rangle]{\langle a \rangle}` to change the arrangement of the  $\langle s \rangle$  series of familiar footnotes.

The value of  $\langle a \rangle$  can be one of the following

- `paragraph` formats all the footnotes of a series as a single paragraph;
- `twocol` formats them as separate paragraphs, but in two columns;
- `threecol`, in three columns.
- `normal`, restore normal arrangement.

You should set up the page layout parameters, and in particular the `\baselineskip` of the footnotes, before you call this macro because its 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.<sup>17</sup>

The notes arrangement must be called after having defined the document geometry setting. If you must change geometry setting inside your document, do not forget to call again `notarrangement`.

`\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 `\hsize` within the document, call `\footparagraph` again afterwards to take account of the new value.

## 6.2 Control line number printing

### 6.2.1 Print line number only at first time

`\Xnumberonlyfirstinline` 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 `\Xnumberonlyfirstinline[\langle s \rangle]`.

<sup>17</sup>There is one tiny proviso about using paragraphed notes: you should not 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. XII.6.3 p. 137 explains why this restriction is necessary.

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

`\Xnumberonlyfirstintwotlines`

`\Xsymlinenum`

Suppose you have a lemma on line 2 and a lemma between line 2 and line 3. With `\Xnumberonlyfirstinline`, the second lemma is considered to be on the same line as the first lemma. But if you use both `\Xnumberonlyfirstinline[⟨s⟩]` and `\Xnumberonlyfirstintwotlines[⟨s⟩]`, the distinction is made. Use `\Xnumberonlyfirstintwotlines[⟨s⟩]` to disable this (⟨s⟩ can be empty if you want to disable it for every series). For setting a particular symbol in place of the line number, you can use `\Xsymlinenum[⟨s⟩]{⟨symbol⟩}` in combination with `\Xnumberonlyfirstinline[⟨s⟩]`. From the second lemma of the same line, the symbol will be used instead of the line number. Note that any command called in `⟨symbol⟩` must be robust. Use `\robustify` to robustify a not robust command.

### 6.2.2 Abbreviate line range

`\Xtwotlines`  
`\Xmorethantwotlines`

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 `\Xtwotlines[⟨s⟩]{⟨text⟩}` and `\Xmorethantwotlines[⟨s⟩]{⟨text⟩}`. The `⟨text⟩` argument of `\Xtwotlines` will be printed if the lemma is on two lines, and the `⟨text⟩` argument of `\Xmorethantwotlines` will be printed if the lemma is on three or more lines. For example:

```
\Xtwotlines{sq.}
\Xmorethantwotlines{sqq.}
```

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

`\Xmorethantwotlines`

If you use `\Xtwotlines` without setting `\Xmorethantwotlines`, the `⟨text⟩` argument of `\Xtwotlines` 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 `\Xtwotlinesbutnotmore[⟨series⟩]`.

It is possible to disable `\Xtwotlinesbutnotmore[⟨series⟩]` with `\Xtwotlinesbutnotmore[⟨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 `\Xendtwotlines` symbol.

`\Xtwotlinesonlyinsamepage`

However, you can force print the final page number with `\Xtwotlinesonlyinsamepage[⟨series⟩]`.

Use `\Xtwotlinesonlyinsamepage[⟨series⟩][false]` to disable this.

You can disable `\Xtwotlines` and related for a specific note by using the ‘[fulllines]’ argument in the note macro cf. 5.2.2 p. 22.

`\Xendtwotlines`

`\Xendmorethantwotlines`

`\Xendtwotlinesbutnotmore`

For endnotes, use `\Xendtwotlines`; `\Xendmorethantwotlines`; `\Xendtwotlinesbutnotmore`; `\Xendtwotlinesonlyinsamepage` instead of `\Xtwotlines`; `\Xmorethantwotlines`; `\Xtwotlinesbutnotmore`; `\Xtwotlinesonlyinsamepage`.

### 6.2.3 Disable line number

`\Xnonumber`

You can use `\Xnonumber[⟨s⟩]` if you do not want to have the line number in a foot-

`\Xendnonnumber` note. To cancel it, use `\Xnonnumber[⟨s⟩][false]`. `\Xendnonnumber[⟨s⟩]` is the same for endnote.

#### 6.2.4 Printing pstart number

`\Xpstart` You can use `\Xpstart[⟨s⟩]` if you want to print the pstart number in the footnote, before the line and subline number. Use `\Xpstart[⟨s⟩][false]` to disable this (⟨s⟩ 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.

`\Xpstarteverytime` By default, the pstart number is printed only in the part of text where you have called `\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 `\Xpstarteverytime[⟨s⟩]`. In this case, the pstart number will be printed every time in footnote.

`\Xonlypstart` In combination with `\Xpstart`, you can use `\Xonlypstart[⟨s⟩]` if you want to print only the pstart number in the footnote, and not the line and subline number. Use `\Xonlypstart[⟨s⟩][false]` to disable this (⟨s⟩ can be empty if you want to disable it for every series).

#### 6.2.5 Space around number

`\Xbeforenumber` With `\Xbeforenumber[⟨s⟩]{⟨l⟩}`, you can add some space before the line number in a footnote. If the line number is not printed, the space is not either. The default value is 0 pt.

`\Xafternumber` With `\Xafternumber[⟨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 0.5 em.

`\Xnonbreakableafternumber` By default, the space defined by `\Xafternumber` is breakable. With `\Xnonbreakableafternumber[⟨s⟩]` it becomes nonbreakable. Use `\Xnonbreakableafternumber[⟨s⟩][false]` to disable this (⟨s⟩ can be empty if you want to disable it for every series).

#### 6.2.6 Space around line symbol

`\Xbeforesymmlinenum` With `\Xbeforesymmlinenum[⟨s⟩]{⟨l⟩}` you can add some space before the line symbol in a footnote. The default value is value set by `\Xbeforenumber`.

`\Xaftersymmlinenum` With `\Xaftersymmlinenum[⟨s⟩]{⟨l⟩}` you can add some space after the line symbol in a footnote. The default value is value set by `\Xafternumber`.

#### 6.2.7 Space in place of number

`\Xinplaceofnumber` If no number or symbolic line number is printed, you can add a space, with `\Xinplaceofnumber[⟨s⟩]{⟨l⟩}`. The default value is 1 em.

`Xendinplaceofnumber` . `\Xendinplaceofnumber[⟨s⟩]{⟨l⟩}` is the same, for critical endnotes.

### 6.2.8 Boxing line number and line symbol

`\Xboxlinenum` 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 `\Xboxlinenum[⟨s⟩]{⟨l⟩}` to do that. To subsequently disable this feature, use `\Xboxlinenum` 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}
\Xafternumber{0em}
\Xboxlinenum{1em}
```

`\Xboxsymlinenum` `\Xboxsymlinenum[⟨s⟩]{⟨l⟩}` is the same as `\Xboxlinenum` but for the line number symbol.

`Xboxlinenumalign` If you put line number in box, it will be aligned left inside the box. However, you can change it using `\Xboxlinenumalign[⟨s⟩]{⟨text⟩}` where `⟨text⟩` can be the following:

**L** to align left (default value);

**R** to align right;

**C** to center.

When using `\Xboxlinenum`, `reledmac` 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.

- `\Xboxstartlinenum[⟨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.
- `\Xboxendlinenum[⟨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.
```

`\Xendboxlinenum` `\Xendboxlinenum[⟨s⟩]{⟨l⟩}`, `\Xendboxlinenumalign[⟨s⟩]{⟨text⟩}`, `\Xendboxstartlinenum[⟨s⟩]{⟨l⟩}`, `\Xendboxendlinenum[⟨s⟩]{⟨l⟩}` are the same as, respectively, `\Xboxlinenum` and `\Xboxlinenumalign`, `\Xboxstartlinenum`, `\Xboxendlinenum` except in endnotes.

## 6.3 Separator between the lemma and the note

### 6.3.1 For footnotes

`\Xlemmaseparator` By default, in a footnote, the separator between the lemma and the note is a right bracket



(\rbracket). You can use `\Xlemmaseparator[\langle s \rangle]{\langle Xlemmaseparator \rangle}` 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.

`\Xbeforelemmaseparator` Using `\Xbeforelemmaseparator[\langle s \rangle]{\langle l \rangle}` 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.

`\Xafterlemmaseparator` Using `\Xafterlemmaseparator[\langle s \rangle]{\langle l \rangle}` you can add some space between separator and note. If your lemma separator is empty, this space will not be printed. The default value is 0.5 em.

`\Xnolemmaseparator` You can suppress the lemma separator, using `\Xnolemmaseparator[\langle s \rangle]`, which is simply a alias of `\Xlemmaseparator[\langle s \rangle]{}`.

`\Xinplaceoflemmaseparator` With `\Xinplaceoflemmaseparator[\langle s \rangle]{\langle l \rangle}` you can add a space if no lemma separator is printed. The default value is 1 em.

### 6.3.2 For endnotes

`\Xendlemmaseparator` By default, there is no separator inside endnotes between the lemma and the content of the note. You can use `\Xendlemmaseparator[\langle s \rangle]{\langle Xendlemmaseparator \rangle}` to change this. The optional argument can be used to specify the series in which it is used. An common value of `\Xendlemmaseparator` 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[\langle s \rangle]{\langle l \rangle}` 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[\langle s \rangle]{\langle l \rangle}` 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.

`\Xendingplaceoflemmaseparator` With `\Xendingplaceoflemmaseparator[\langle s \rangle]{\langle l \rangle}` you can add some space if you chose to remove the lemma separator. The default value is 0.5 em.

## 6.4 Font style

### 6.4.1 For line number

`\Xnotenumfont` `\Xnotenumfont[\langle s \rangle]{\langle command \rangle}` is used to change the font style for line numbers in critical footnotes ; `\langle command \rangle` must be one (or more) switching command, like `\bfseries`.

`\Xendnotenumfont` `\Xendnotenumfont[\langle s \rangle]{\langle command \rangle}` is used to change the font style for line numbers in critical footnotes. `\langle command \rangle` must be one (or more) switching command, like `\bfseries`.

`\notenumfontX` `\notenumfontX[\langle s \rangle]{\langle command \rangle}` is used to change the font style for note numbers in familiar footnotes. `\langle command \rangle` must be one (or more) switching command, like `\bfseries`.

`\Xnotefontsize` `\Xnotefontsize[\langle s \rangle]{\langle command \rangle}` is used to define the font size of critical foot-

	notes of the series. The default value is <code>\footnotesize</code> . The <i>command</i> must not be a size in pt, but a standard $\TeX$ size, like <code>\small</code> .
<code>\notefontsizeX</code>	<code>\notefontsizeX[<i>s</i>]{<i>command</i>}</code> is used to define the font size of critical footnotes of the series. The default value is <code>\footnotesize</code> . The <i>command</i> must not be a size in pt, but a standard $\TeX$ size, like <code>\small</code> .
<code>\Xendnotefontsize</code>	<code>\Xendnotefontsize[<i>s</i>]{<i>l</i>}</code> is used to define the font size of end critical footnotes of the series. The default value is <code>\footnotesize</code> . The <i>command</i> must not be a size in pt, but a standard $\TeX$ size, like <code>\small</code> .

### 6.4.2 For the lemma

<code>\Xlemmadisablefontselection</code>	By default, font of the lemma in footnote is the same as font of the lemma in the main text. For example, if the lemma is in italic in the main text, it is also in italic in note. The <code>\Xlemmadisablefontselection[<i>s</i>]</code> command allows to disable it for a specific series.
<code>\Xendlemmadisablefontselection</code>	By default, font of the lemma in endnote is the same as font of the lemma in the main text. For example, if the lemma is in italic in the main text, it is also in italic in note. The command allows <code>\Xendlemmadisablefontselection[<i>s</i>]</code> to disable it for a specific series.

## 6.5 Styles of notes content

<code>\Xparindent</code>	By default, reledmac does not add indentation before the paragraphs inside critical footnotes. Use <code>\Xparindent[<i>series</i>]</code> to enable indentation.
<code>\parindentX</code>	By default, reledmac does not add indentation before the paragraphs inside familiar footnotes. Use <code>\parindentX[<i>series</i>]</code> to enable indentation.
<code>\Xhangindent</code>	For critical notes NOT paragraphed you can define an indent with <code>\Xhangindent[<i>s</i>]{<i>l</i>}</code> , 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.
<code>\hangindentX</code>	For familiar notes NOT paragraphed you can define an indentation with <code>\Xhangindent[<i>s</i>]{<i>l</i>}</code> , 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.

## 6.6 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 :

```
\Xbhooknote{\renewcommand{\thepstart}{\arabic{pstart}}.}}
```

<code>\Xbhooknote</code>	<code>\Xbhooknote[<i>series</i>]{<i>code</i>}</code> is to be used at the beginning of the critical footnotes.
<code>\bhooknoteX</code>	<code>\bhooknoteX[<i>series</i>]{<i>code</i>}</code> is to be used at the beginning of the familiar footnotes.
<code>\Xendbhooknote</code>	<code>\Xendbhooknote[<i>series</i>]{<i>code</i>}</code> is to be used at the beginning of the endnotes.

## 6.7 Options for footnotes in columns

### 6.7.1 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  $\text{\LaTeX}$ . You can also let `<code>` empty.

`\raggedright` to have text left aligned, but *without hyphenation*. That is the default `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*.

### 6.7.2 Size of the columns

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

<code>\Xhsizetwocol</code>	<code>\Xhsizetwocol[⟨s⟩]{⟨l⟩}</code> is used to change width of a column when critical notes are displaying in two columns. Default value is <code>.45 \hspace</code> .
<code>\Xhsizethreecol</code>	<code>\Xhsizethreecol[⟨s⟩]{⟨l⟩}</code> is used to change width of a column when critical notes are displaying in three columns. Default value is <code>.3 \hspace</code> .
<code>\hsizetwocolX</code>	<code>\hsizetwocol[⟨s⟩]{⟨l⟩}</code> is used to change width of a column when familiar notes are displaying in two columns. Default value is <code>.45 \hspace</code> .
<code>\hsizethreecolX</code>	<code>\hsizethreecolX[⟨s⟩]{⟨l⟩}</code> is used to change width of a column when familiar notes are displaying in three columns. Default value is <code>.3 \hspace</code> .

## 6.8 Options for paragraphed footnotes

### 6.8.1 Mark separation of notes

<code>\Xafternote</code>	You can add some space after a note by using <code>\Xafternote[⟨s⟩]{⟨l⟩}</code> (for critical footnotes) or <code>\afternoteX[⟨s⟩]{⟨l⟩}</code> (for familiar footnotes). The default value is <code>1em plus.4em minus.4em</code> .
<code>\afternoteX</code>	
<code>\Xparafootsep</code>	For paragraphed footnotes (see below), you can choose the separator between each note by using <code>\Xparafootsep[⟨s⟩]{⟨text⟩}</code> for critical notes and <code>\parafootsepX</code> for familiar notes. A common separator is the double pipe ( <code>  </code> ), which you can set by using <code>\parafootsep{\$\parallel\$}</code> .
<code>\parafootsepX</code>	

Note that if the symbol defined by `\Xsymlinenum` must be used at the beginning of a note, the `\Xparafootsep` / `\parafootsepX` is not used before this note.

### 6.8.2 Ragging

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

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

## 6.9 Options for block of notes

### 6.9.1 Text before notes

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

### 6.9.2 Spacing

`\Xbeforenotes` You can change the vertical space printed before the rule of the critical notes with `\Xbeforenotes[⟨s⟩]{⟨l⟩}`. The default value is 1.2em plus .6em minus .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 .6em minus .6em.

**Be careful, the standard  $\LaTeX$  footnote rule, which is used by `eledmac`, decreases 3pt. These 3pt 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 want 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.

`\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 want 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.

### 6.9.3 Rule

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

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

#### 6.9.4 Maximum height

`\Xmaxhnotes` 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 `\Xmaxhnotes[ $\langle s \rangle$ ]{ $\langle l \rangle$ }`. Be careful: the length can't be flexible, and is relative to the the current font. For example, if you want the note to take, at most, 33 of the text height, do `\Xmaxhnotes{.33\textheight}`.

`\maxhnotesX` `\maxhnotesX[ $\langle s \rangle$ ]{ $\langle l \rangle$ }` 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 ca not be broken between two pages, even if you used these commands. The debug is in the `todolist`.

### 6.10 Footnotes and the `reledpar` columns

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

`\noteswidthliketwocolumnsX`

- `\Xnoteswidthliketwocolumns[ $\langle s \rangle$ ]` to create critical notes with a two-column size width. Use `\Xnoteswidthliketwocolumns[ $\langle s \rangle$ ] [false]` to disable it.
- `\noteswidthliketwocolumnsX[ $\langle s \rangle$ ]` to create familiar notes with a two-column size width. Use `\noteswidthliketwocolumnsX[ $\langle s \rangle$ ] [false]` to disable it.

### 6.11 Endnotes in one paragraph

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

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

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

## 7 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 `reledmac` macros relating to fonts that are intended for manipulation by the user: `\endashchar`, `\fullstop`, `\numlabfont`, and `\rbracket`.

`\numlabfont` 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.

`\endashchar`  
`\fullstop`  
`\rbracket`

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 T<sub>E</sub>X 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 reledmac’s standard style). For polyglossia, when the lemma is RTL, the bracket automatically switches to a left bracket.

`\select@lemmafont`

We will briefly discuss `\select@lemmafont` 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 `\edtext` 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@lemmafont` does the work of decoding reledmac’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@lemmafont` 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@lemmafont` selects the appropriate font for the note using that font specifier.

reledmac uses `\select@lemmafont` 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.

## 8 Verse

### 8.1 Basic

`\stanza` Use `\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.

### 8.2 Define stanza indents

`\stanzaindentbase` Lines within a stanza may be indented. The indents are integer multiples of the length

`\stanzaindentbase`, whose default value is 20pt.

`\setstanzaindents`

In order to use the stanza macros, **one must set the indentation values**. First the value of `\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

`\setstanzaindents{3,1,2,1,2}`.

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 `\sethanginsymbol:` see p. 8.6 p. 40.

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.

### 8.3 Repeating stanza indents

Since version 0.13, if the indentation is repeated every  $n$  verses of the stanza, you can define only the  $n$  first indentations, and say they are repeated, defining the value of the `stanzaindentrepetition` counter at  $n$ . For example:

```
\setstanzaindents{5,1,0}
\setcounter{stanzaindentrepetition}{2}
```

is like

```
\setstanzaindents{0,1,0,1,0,1,0,1,0,1,0}
```

**Be careful: the feature is changed in eledmac 1.5.1. See Appendix A.3 p. 260.**

If you don't use the `stanzaindentrepetition` counter, make sure you have at least one more numerical entry in `\setstanzavalues` than the number of lines in the stanza.

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

```
\setcounter{stanzaindentrepetition}{0}
```

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.

## 8.4 Manual stanza indent

`\stanzaindent` You can set the indent of some specific verse by calling `\stanzaindent{⟨value⟩}` at the beginning of the verse, before any other character. In this case, the indent defined by `\setstanzaindent*` for this verse is skipped, and `{⟨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`.

## 8.5 Stanza breaking

`\setstanzapenalties` 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. 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.

## 8.6 Hanging symbol

`\sethangingsymbol` It is possible to insert a symbol in each line of hanging verse, as in French typography for ‘[’. To insert it in `reledmac`, use macro `\hangingsymbol⟨h⟩` with this code. In the example of French typography, do

```
\sethangingsymbol{[\,]}
```

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

```
\sethangingsymbol{\protect\hfill}
```



## 8.7 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 16.2 p. 53 for further details.

## 8.8 Hanging symbol

It is possible to insert a symbol on each line of hanging verse, as in French typography for ‘[’. To insert in `reledmac`, use macro `\sethangingsymbol{⟨symbol⟩}`. Fro exemple:

```
\hangingsymbol{[,}
```

## 8.9 Content before/after verses

It is possible to add content, like a subtitle or a spacing, 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.

## 8.10 Various tools

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

`\flagstanza` Putting `\flagstanza[⟨len⟩]{⟨text⟩}` at the start of a line in a stanza (or elsewhere) will typeset `⟨text⟩` at a distance `⟨len⟩` before the line. The default `⟨len⟩` 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{\numberit}{%
  \refstepcounter{stanzanum}%
  \flagstanza{\thestanzanum}%
}
...
\stanza[\numberit]
\numberit First line...&
  rest of stanza\&

\stanza[\numberit]
First line, second stanza...
```

## 9 Grouping

In a `minipage` environment  $\TeX$  changes `\footnote` numbering from arabic to alphabetic and puts the footnotes at the end of the `minipage`.

`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 5.4) in a `minipage` but unlike with `\footnote` the numbering scheme is unaltered.

`ledgroup` Minipages, of course, are not broken across pages. Footnotes in a `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.

`ledgroupsize` The `ledgroupsize` environment is similar to `ledgroup` except that you must specify a width for the environment, as with a `minipage`.  
`\begin{ledgroupsize}[\langle pos \rangle]{\langle width \rangle}`.

The required `\langle width \rangle` argument is the text width for the environment. The optional `\langle pos \rangle` argument is for positioning numbered text within the normal `textwidth`. It may be one of the characters:

l (left) numbered text is flush left with respect to the normal `textwidth`. This is the default.

c (center) numbered text is in the center of the `textwidth`.

r (right) numbered text is flush right with respect to the normal `textwidth`.

Note that normal text, footnotes, and so forth are all flush left.

`\begin{ledgroupsize}\textwidth` is effectively the same as `\begin{ledgroup}`

## 10 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.

### 10.1 Basic use

`\edlabel` First you place a label in the text using the command `\edlabel{\langle lab \rangle}`. `\langle lab \rangle` can be almost anything you like, including letters, numbers, punctuation, or a combination—anything but spaces; you might say `\edlabel{toves-3}`, for example.<sup>18</sup>

`\edpageref` Elsewhere in the text, either before or after the `\edlabel`, you can refer to its location via `\edpageref{\langle lab \rangle}`, or `\edlineref{\langle lab \rangle}` will produce, respectively, the page, line, sub-line and `pstart` on which the `\edlabel{\langle lab \rangle}` command occurred.

`\sublineref` 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`

<sup>18</sup>More precisely, you should stick to characters in the  $\TeX$  categories of “letter” and “other”.

`\pstartref` commands can also be used in the apparatus to refer to `\edlabels` in the text.

The `\edlabel` command works by writing macros to `ℒTeX.aux` file. You will need to process your document through `ℒTeX` 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.)

## 10.2 Refer to a critical notes

If you want to refer to a word inside an `\edtext{⟨lemma⟩}{⟨app⟩}` 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\emph{n}}
```

## 10.3 Cross-referencing which return a number in any case

`\xpageref` Where `#1` stands for the reference.

`\xlineref` However, there are situations in which you will want `reledmac` to return a number without displaying any warning messages about undefined labels or the like: if you want  
`\xsublineref` to use the reference in a context where `ℒTeX` is looking for a number, such a warning  
`\xpstartref` will lead to a complaint that the number is missing. This is the case for references used within the argument to `\linenum`, for example (see 5.2.5 p. 23).

For this situation, four variants of the reference commands, with the `x` prefix, are supplied: `\xpageref`, `\xlineref`, `\xsublineref` 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 `\edlabel{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 will not be added. (Indeed, it is not a limitation, but a feature.)

### 10.3.1 Cross-referencing in order to define line number of a critical note

`\xxref` 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., 5.2.5 p. 23 above) and sets the beginning page, line and subline numbers to those of the place where `\edlabel{mouse}` was placed, and the ending numbers to those where `\edlabel{elephant}` occurs.

## 10.4 Not automatic cross-referencing

`\edmakelabel` 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.

## 10.5 Normal L<sup>A</sup>T<sub>E</sub>X cross-referencing

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

## 10.6 References to lines commented in the apparatus

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

`\applabel` If you use `\applabel{<label>}` inside the second argument of a `\edtext`, `reledmac` 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 `<label>:start`, while the label at the end will have the title `<label>:end`.

If you use `\linenum` (5.2.5 p. 23) to refer to these labels, `reledmac` will use your line settings to refer to the passage.

`\appref` You can also use `\appref{<label>}` and `\apprefwithpage{<label>}` to refer to these  
`\apprefwithpage` 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.

`\setapprefprefixsingle` If you use `\setapprefprefixsingle{<prefix>}`, `<prefix>` will be printed before the  
`\setapprefprefixmore` line numbers of a `\appref`-reference. If you use `\setapprefprefixmore{<prefix>}`, `<prefix>` will be printed before the line numbers, if you refer to more than one line.

For example, you may use:

```
\setapprefprefixsingle{line~}
```

`\setapprefprefixmore{lines~}`

Note that if you have not used `\setapprefprefixmore` is empty, argument of `\setapprefprefixsingle` will be used in any case.

`\Xtwolinesappref`  
`\Xmorethantwolinesappref`  
`\Xtwolinesbutnotmoreappref`  
`\Xtwolinesonlyinsamepage`

If you use `\Xtwolines`, `\Xmorethantwolines`, `\Xtwolinesbutnotmore` and/or `\Xtwolinesonlyinsamepage` (6.2.2 p. 30) *without the optional series argument*, the setting will also be available for `\appref`.

The commands `\Xtwolinesappref{<text>}`, `\Xmorethantwolinesappref{<text>}`, `\Xtwolinesbutnotmoreappref` `\Xtwolinesonlyinsamepageappref` 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>}`.

`\Xendtwolinesapprefwithpage`  
`\Xendmorethantwolinesapprefwithpage`  
`\Xendtwolinesbutnotmoreapprefwithpage`  
`\Xendtwolinesonlyinsamepageapprefwithpage`

If you use one of `\Xendtwolines`, `\Xendmorethantwolines`, `\Xendtwolinesbutnotmore`, `\Xendtwolinesonlyinsamepage` (6.2.2 p. 30) *without the optional series argument*, the setting will also be available for `\apprefwithpage`.

The commands `\Xendtwolinesappref{<text>}`, `\Xendmorethantwolinesappref{<text>}`, `\Xendtwolinesbutnotmoreappref`, `\Xendtwolinesonlyinsamepageappref` 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>}`.

## 11 Side notes

### 11.1 Basis

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

`\ledinnernote`  
`\ledouternote`

`\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`  
`\sidenotemargin`

`\ledsidenote{<text>}` will put `<text>` into the margin specified by the current setting of `\sidenotemargin{<location>}`. The permissible value for `<location>` is one out of the list `left`, `right`, `inner`, or `outer`, for example `\sidenotemargin{outer}`. The package's default setting is `\sidenotemargin{right}`

to typeset `\ledsidenotes` in the right hand margin. This is the opposite to the default margin for line numbers. The style for a `\ledsidenote` follows that for a `\ledleftnote` or a `\ledrightnote` depending on the margin it is put in.

If two, say, `\ledleftnote`, commands are called in the same line the second `<text>` will obliterate the first. There is no problem though with having both a left and a right sidenote on the same line.

## 11.2 Setting

### 11.2.1 Width

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

### 11.2.2 Vertical position

`\rightnoteupfalse` By default, sidenotes are placed to align with the last line of the note to which it refers.  
`\leftnoteupfalse` If you want they 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).

### 11.2.3 Distance to the main text

`\ledlsnotesep` The texts are put a distance `\ledlsnotesep` (or `\ledrsnotesep`) into the left (or right)  
`\ledrsnotesep` margin. These lengths are initially set to the value of `\linenumsep`.  
`\ledlsnotefontsetup` These macros specify how the sidenote texts are to be typeset. The initial definitions  
`\ledrsnotefontsetup` are:

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

These can of course be changed to suit.

### 11.2.4 Separator between notes

`\setsidenotesep` 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 use `\setsidenotesep{<sep>}`.

## 12 Indexing

### 12.1 Basis

`\edindex`  $\TeX$  provides the `\index{<item>}` command for specifying that `<item>` and the current  
 page number should be added to the raw index (`idx`) file. The `\edindex{<item>}` macro  
 can be used in numbered text to specify that `<item>` and the current page & line number  
 should be added to the raw index file.

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

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

The minimal version of 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.

## 12.2 Separator between page and line numbers

`\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<sup>19</sup>.

```
page_compositor ":"
delim_r ":"
```

Read the MAKEINDEX program's handbook about the .ist file.

## 12.3 Using xindy

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

pagenumber-linenumber

An example of such a file is provided in the "examples" folder. Read the xindy handbook to learn how to use it.<sup>20</sup>

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

In any case, you must load reledmac with the 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 `reledmac-markup-attr.xdy`, but you can change it by using your own as an argument of the `xindy+hyperref` option.

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

<sup>19</sup>For further detail, you can read <http://tex.stackexchange.com/a/32783/7712>.

<sup>20</sup>Or, for people who read French, read <http://geekographie.maieul.net/174>.

1. Use `xindy+hyperref` option when loading the `reledmac` 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 `\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.<sup>21</sup>.

## 12.4 Advanced setting

`\edindexlab` 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.

## 13 Tabular material

$\TeX$ ’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, `reledmac` provides some simple tabulation environments that can be line numbered. The environments can also be used in normal unnumbered text.

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

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

1	2	3
a	bb	ccc
AAA	BB	C

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.

<sup>21</sup>These are the recommended lines to provide the best possible compatibility between `hyperref` and `xindy`, even without using `reledmac`.





1	2	3	4	5
Q		fd	h	qwertziohg
v	wptz	x	y	vb
g	nnn			
		pq dgh		
k		1	co	ghweropjklmnbvcxys
1	2	3		

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

[illegible]

1	2	3	4
$a$	$\underbrace{\hspace{1.5cm}}$		$d$
$A$	$B$	$C$	$D$

```
\begin{edarrayc}
& 1 & 2 & 3 & \\
& 4 & 5 & 6 & \\
\edatleft[left =]{\}{1.5\baselineskip}
& 7 & 8 & 9 & \\
\edatright[= right]{\}{1.5\baselineskip}
\end{edarrayc}
```

$$left = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix} = right$$

`\edbeforetab`      `\edbeforetab{⟨text⟩}{⟨entry⟩}`, where `⟨entry⟩` is an entry in the leftmost column, typesets `⟨text⟩` left justified before the `⟨entry⟩`. Similarly `\edaftertab{⟨entry⟩}{⟨text⟩}`, where `⟨entry⟩` is an entry in the rightmost column, typesets `⟨text⟩` right justified after the `⟨entry⟩`.

For example:

```
\begin{edarrayl}
      A & 1 & 2 & 3 \\
\edbeforetab{Before}{B} & 1 & 3 & 6 \\
      C & 1 & 4 & \edaftertab{8}{After} \\
      D & 1 & 5 & 0 \\
\end{edarrayl}
```

	$\begin{array}{rrrr} A & 1 & 2 & 3 \\ B & 1 & 3 & 6 \\ C & 1 & 4 & 8 \\ D & 1 & 5 & 0 \end{array}$		After
Before			

`\edvertline`      The macro `\edvertline{⟨height⟩}` draws a vertical line `⟨height⟩` high (contrast this with `\edatright` where the size argument is half the desired height).

`\edvertdots`

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

$a$	$b$	$C$	$d$	
$v$	$w$	$x$	$y$	
$m$	$n$	$o$	$p$	
$k$		$L$	$cvb$	

The `\edvertdots` macro is similar to `\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 (`\chapter`, `\section` etc.) can be used inside numbered text. In this case, you must call them as an optional argument of `\pstart` (4.2.3 p. 16):

```
\pstart[\section{section}]
Pstart content.
\pend
```

The line which contains them will not be numbered, and you can not add critical notes inside.

## 14.2 Sectioning commands with line numbering and critical notes

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`:

```
\pstart
\eledsection*{xxxx\ledsidenote{section}}
\pend
\pstart
\eledsubsection*{xxxx\ledsidenote{sub}}
\pend
\pstart
normal text
\pend
```

At the first run, you will see only the text. It is 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 not be added automatically. You have to insert it manually via `\beforeeledchapter`, which must be called outside of a numbered section.

## 14.3 Optimization

`\noeledsec` If you are not going to have any `\eledxxx` commands, then load `reledmac` with `\noeledsec` option. That will suppress the generation of unneeded `.eledsec` file, keep memory and make `reledmac` 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 do not 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

### 16.1 Control page breaking

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

`\ledpb`  
`\lednopb`

- `\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 `\ledpbsetting{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.  $n + 1$ .

`\ledpbsetting`

If you are using `reledpar` 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.

### 16.2 Prevent page break in a long verses

`\lednopbinversettrue` You can also decide to prevent page breaks between two lines of a long verse. To do this, use `nopbinverse` when loading package, or add `\lednopbinversettrue` 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 `\ledpbsetting{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

<code>\extensionchars</code>	When the package assembles the name of the auxiliary file for a section, it prefixes <code>\extensionchars</code> 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 <code>\renewcommand{\extensionchars}{!}</code> , then you would get temporary files called <code>jobname. !1</code> , <code>jobname. !2</code> , etc.
<code>\ifledfinal</code>	The package can take options. The option ‘final’, which is the default is for final typesetting; this sets <code>\ifledfinal</code> to TRUE. The other option, ‘draft’, may be useful during earlier stages and sets <code>\ifledfinal</code> to FALSE.
<code>\showlemma</code>	<p>The lemma within the text is printed via <code>\showlemma{lemma}</code>. Normally, or with the ‘final’ option, the definition of <code>\showlemma</code> is:</p> <pre>\newcommand*{\showlemma}[1]{#1}</pre> <p>so it just produces its argument. With the ‘draft’ option it is defined as</p> <pre>\newcommand*{\showlemma}[1]{\textit{#1}}</pre> <p>so that its argument is typeset in an italic font, which may make it easier to check that all lemmas have been treated.</p> <p>If you would prefer some other style, you could put something like this in the preamble:</p> <pre>\ifledfinal\else \renewcommand{\showlemma}[1]{\textbf{#1}}% or simply ... [1]{#1} \fi</pre>

### 17.1 Known and suspected limitations

#### 17.2 ‘No room for a new’

Sometime, especially when using `reledmac` with other packages, you could obtain warning message such ‘no room for a new count’ or ‘no room for a new write’.

The first thing in order to prevent such problem is to use the options to optimize `reledmac`. For example, if you need only two series of notes, use `series={A,B}` option. Read ?? p. ?? in order to know which are there options.

However, if with these options you still have such message, here are some tricks.

‘**no room for a new count**’ is often caused by a conjunction with `biblatex`. Load `reledmac` (and `reledpar`) *before* `biblatex`.

‘**no room for a new write**’ can be caused by with multiple indexes. In this case, use `indextools` of `imakeidx` with the `splitindex` option, in order to obtain only one `.idx` file. If that does not solve your problem, you can use `morewrites` package. That should solve the problem, but  $\LaTeX$  will be slower.

If after reading and applying these advices you have still problem, contact us with a minimal working example.

### 17.2.1 Marginal notes

In general, reledmac’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.

### 17.2.2 Paragraph shape

`\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, reledmac 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 does not crop up at all often.

### 17.2.3 Paragraphed footnotes

The restriction on explicit line-breaking in paragraphed footnotes, mentioned in a footnote 17 p. 29, and described in more detail on XII.6.3 p. 136, 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.

`\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 68) to increase the estimate. You have to use `\renewcommand` for this, like:  
`\renewcommand{\footfudgefiddle}{68}`

### 17.2.4 Use with other packages

Because of reledmac’s complexity it may not play well with other packages. In particular reledmac is sensitive to commands in the arguments to the `\edtext` and `\*footnote` macros (this is discussed in more detail in section VI, and in particular the discussion about `\no@expands` and `\morenoexpands`). You will have to see what works or doesn’t work in your particular case.

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

```
... \edtext{\colorbox{mycolor}{lemma}}{\Afootnote{...\colorbox{...}}}
```

If you actually try this<sup>22</sup> you will find L<sup>A</sup>T<sub>E</sub>X whinging ‘Missing { inserted’, and then things start to fall apart. The trick in this case is to specify either:

```
\newcommand{\morenoexpands}{\let\colorbox=0}
```

or

```
\makeatletter
\newcommand{\morenoexpands}{\let\colorbox\@secondoftwo}
\makeatother
```

(\@secondoftwo is an internal L<sup>A</sup>T<sub>E</sub>X 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 \textcolor instead, like

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

there is no need to fiddle with \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:

```
\makeatletter
\newcommand{\morenoexpands}{\let\textcolor\@secondoftwo}
\makeatother
```

It took Peter Wilson 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.

If you want to use the option *bottom* of the footmisc package, you must load this package *before* the reledmac package.

### 17.3 Parallel typesetting

Peter Wilson has developed the ledpar package as an extension to ledmac specifically for parallel typesetting of critical texts. This also cooperates with the babel / polyglossia packages for typesetting in multiple languages. reledpar is the successor of the primitive ledpar package.

Peter Wilson also developed the 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 Xe(La)TeX

---

<sup>22</sup>Reported by Dirk-Jan Dekker in the CTT thread ‘Incompatibility of “color” package’ on 2003/08/28.



## I Implementation overview

We present the `reledmac` code in roughly the order in which it is used during a run of  $\TeX$ . The order is *exactly* that in which it is 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.

After package options, we begin with the commands you use to start and stop line numbering in a section of text (Section II). 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 V); 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 VI), followed by the macros that take each paragraph apart, attach the line numbers and insertions, and send the result to the vertical list (Section VII). The footnote commands (Section XII) and output routine (Section ??) finish the main part of the processing; cross-referencing (Section XXIII) and endnotes (Section XIX) complete the story.

In what follows, macros with an `@` in their name are more internal to the workings of `reledmac` 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.

## II Preliminaries

### II.1 Links with original `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.

### II.2 Package declaration

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

```
1 \code
2 \NeedsTeXFormat{LaTeX2e}
3 \ProvidesPackage{reledmac}[2015/07/19 v2.0.0 typeset critical edition]
```

### II.3 Package options

```

\ifledfinal Use this to remember which option is used, set and execute the options with final as the
\ifnocritical@ default. We use xkeyval in order to manage options with argument.
\if@noeled@sec 4 \RequirePackage{xkeyval}
\ifnoend@
\ifnofamiliar@ The parledgroup option is for reledpar. However, it has consequence on reledmac
\ifnoledgroup@ internal command. So we need to define the boolean now.
\ifparapparatus@ 5 \newif\ifparledgroup
\ifnoquotation@ And now, the options of reledmac.
\iflednopbinverse 6 \DeclareOptionX{series}[A,B,C,D,E]{\xdef\default@series{#1}}
\ifparledgroup 7 \ExecuteOptionsX{series}%
\ifwidthliketwocolumns 8
\ifxindy@ 9 \newif\if@noeled@sec%
\ifxindyhyperref@ 10 \DeclareOptionX{noeledsec}{\@noeled@sectrue}
\ifeledmaccompat@ 11
12 \newif\ifnocritical@%
13 \DeclareOptionX{nocritical}{\nocritical@true}%
14
15 \newif\ifnofamiliar@%
16 \DeclareOptionX{nofamiliar}{\nofamiliar@true}%
17
18 \newif\ifnoledgroup@%
19 \DeclareOptionX{noledgroup}{\noledgroup@true}%
20
21 \newif\ifnoend@%
22 \DeclareOptionX{noend}{%
23 \let\l@dend@open\@gobble%
24 \let\l@dend@close\relax%
25 \global\let\l@dend@stuff=\relax%
26 \noend@true%
27 }%
28
29 \newif\ifnoquotation@
30 \DeclareOptionX{noquotation}{\noquotation@true}
31
32
33 \newif\ifledfinal
34 \DeclareOptionX{final}{\ledfinaltrue}
35 \DeclareOptionX{draft}{\ledfinalfalse}
36 \ExecuteOptionsX{final}
37
38 \newif\ifparapparatus@
39 \DeclareOptionX{parapparatus}{\parapparatus@true}
40
41 \newif\iflednopbinverse
42 \DeclareOptionX{nopbinverse}{\lednopbinversetrue}
43
44 \newif\ifwidthliketwocolumns%

```

```

45 \DeclareOptionX{widthliketwocolumns}{\widthliketwocolumnstrue}%
46
47 \newif\ifxindy@
48 \DeclareOptionX{xindy}[eledmac-markup-attr.xdy]{%
49   \AtBeginDocument{\immediate\openout\eledmac@xindy@out=#1}%
50   \newwrite\eledmac@xindy@out%
51   \xindy@true%
52   \gdef\eledmacmarkuplocdepth{:depth 1}%
53   \AtEndDocument{\immediate\closeout\eledmac@xindy@out}%
54 }%
55
56 \newif\ifxindyhyperref@
57 \DeclareOptionX{xindy+hyperref}{%
58   \xindyhyperref@true%
59 }%
60
61 \newif\ifeledmaccompat@%
62 \DeclareOptionX{eledmac-compat}{%
63   \eledmaccompat@true%
64 }%

```

We use the starred form of `\ProcessOptionsX` 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 `ctt` thread *Class/package option processing*, on 27 February 2004.

```

65 \ProcessOptionsX*\relax
66

```

## II.4 Loading packages

Loading package xargs to declare commands with optional arguments. Etoolbox is also used to make code clearer - for example, in dynamic command names (which can replace `\csname` etc.). Use suffix to declare commands with a starred version, xstring to work with strings, ifluatex and ifxetex to test if Lua<sub>TeX</sub> or X<sub>Y</sub>TeX is running, and ragged2e to manage ragged for paragraphed notes.

```

67 \RequirePackage{xargs}
68 \RequirePackage{etoolbox}
69 \RequirePackage{etex}
70 \reserveinserts{32}
71 \RequirePackage{suffix}
72 \RequirePackage{xstring}
73 \RequirePackage{ifluatex}
74 \RequirePackage{ragged2e}
75 \RequirePackage{ragged2e}
76 \RequirePackage{ifxetex}%

```

## II.5 Boolean flags

`\ifl@dmemoir` Define a flag for if the memoir class has been used.

```

77 \newif\ifl@dmemoir
78 \ifclassloaded{memoir}{\l@dmemoirtrue}{\l@dmemoirfalse}
79

\ifl@imakeidx Define a flag for if the imakeidx package has been used.
80 \newif\ifl@imakeidx
81 \ifpackageloaded{imakeidx}{\l@imakeidxtrue}{}%False is the default value

\ifl@indextools Define a flag for if the indextools package has been used.
82 \newif\ifl@indextools%
83 \ifpackageloaded{indextools}{%
84   \l@indextoolstrue%
85   \l@imakeidxtrue%
86   \let\imki@wrindexentry\indtl@wrindexentry%
87   }{}%

False is the default value. We consider indextools as a variant of imakeidx. That is why
we set \ifl@imakeidx to true. We also let \imki@wrindexentry to \indtl@wrindexentry.

\if@RTL The \if@RTL is defined by the bidi package, which is sometimes loaded by polyglossia.
But we define it as well if the bidi package is not loaded.
88 \ifdef{\if@RTL}{\newif\if@RTL}

```

## II.6 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.

```

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

\reledmac@error Write an error message.
90 \newcommand{\reledmac@error}[2]{\PackageError{reledmac}{#1}{#2}}

\led@err@NumberingStarted
\led@err@NumberingNotStarted
\led@err@NumberingShouldHaveStarted
91 \newcommand*{\led@err@NumberingStarted}{%
92   \reledmac@error{Numbering has already been started}{\@ehc}}
93 \newcommand*{\led@err@NumberingNotStarted}{%
94   \reledmac@error{Numbering was not started}{\@ehc}}
95 \newcommand*{\led@err@NumberingShouldHaveStarted}{%
96   \reledmac@error{Numbering should already have been started}{\@ehc}}

\led@err@edtextoutsidepstart
97 \newcommand*{\led@err@edtextoutsidepstart}{%
98   \reledmac@error{\string\edtext\space outside numbered paragraph (\pstart\ldots\pend)}{\@ehc}}

\led@mess@NotesChanged
99 \newcommand*{\led@mess@NotesChanged}{%
100   \typeout{reledmac reminder: }}%

```

```

101 \typeout{ The number of the footnotes in this section
102         has changed since the last run.}%
103 \typeout{ You will need to run LaTeX two more times
104         before the footnote placement}%
105 \typeout{ and line numbering in this section are
106         correct.}}

\led@mess@SectionContinued

107 \newcommand*{\led@mess@SectionContinued}[1]{%
108 \message{Section #1 (continuing the previous section)}}

\led@err@LineationInNumbered

109 \newcommand*{\led@err@LineationInNumbered}{%
110 \reledmac@error{You can't use \string\lineation\space within
111                a numbered section}\@ehc}}

\led@warn@BadLineation
\led@warn@BadLinenummargin
\led@warn@BadLockdisp
\led@warn@BadSublockdisp

112 \newcommand*{\led@warn@BadLineation}{%
113 \reledmac@warning{Bad \string\lineation\space argument}}
114 \newcommand*{\led@warn@BadLinenummargin}{%
115 \reledmac@warning{Bad \string\linenummargin\space argument}}
116 \newcommand*{\led@warn@BadLockdisp}{%
117 \reledmac@warning{Bad \string\lockdisp\space argument}}
118 \newcommand*{\led@warn@BadSublockdisp}{%
119 \reledmac@warning{Bad \string\sublockdisp\space argument}}

\led@warn@NoLineFile

120 \newcommand*{\led@warn@NoLineFile}[1]{%
121 \reledmac@warning{Can't find line-list file #1}}

\led@warn@LineFileObsolete

122 \newcommand*{\led@warn@Obsolete}[1]{%
123 \reledmac@warning{Line-list file #1 was obsolete. We have not read it. Please run LaTeX again.}}

\led@warn@BadAdvancelineSubline
\led@warn@BadAdvancelineLine

124 \newcommand*{\led@warn@BadAdvancelineSubline}{%
125 \reledmac@warning{\string\advanceline\space produced a sub-line
126                  number less than zero.}}
127 \newcommand*{\led@warn@BadAdvancelineLine}{%
128 \reledmac@warning{\string\advanceline\space produced a line
129                  number less than zero.}}

\led@warn@BadSetline
\led@warn@BadSetlinenum

130 \newcommand*{\led@warn@BadSetline}{%
131 \reledmac@warning{Bad \string\setline\space argument}}
132 \newcommand*{\led@warn@BadSetlinenum}{%
133 \reledmac@warning{Bad \string\setlinenum\space argument}}

```

```

\led@err@PstartNotNumbered
\led@err@PstartInPstart 134 \newcommand*{\led@err@PstartNotNumbered}{%
\led@err@PendNotNumbered 135 \reledmac@error{\string\pstart\space must be used within a
\led@err@PendNoPstart 136 numbered section}{\@ehc}}
\led@err@AutoparNotNumbered 137 \newcommand*{\led@err@PstartInPstart}{%
\led@err@NumberingWithoutPstart 138 \reledmac@error{\string\pstart\space encountered while another
139 \string\pstart\space was in effect}{\@ehc}}
140 \newcommand*{\led@err@PendNotNumbered}{%
141 \reledmac@error{\string\pend\space must be used within a
142 numbered section}{\@ehc}}
143 \newcommand*{\led@err@PendNoPstart}{%
144 \reledmac@error{\string\pend\space must follow a \string\pstart}{\@ehc}}
145 \newcommand*{\led@err@AutoparNotNumbered}{%
146 \reledmac@error{\string\autopar\space must be used within a
147 numbered section}{\@ehc}}
148 \newcommand*{\led@err@NumberingWithoutPstart}{%
149 \reledmac@error{\string\beginnumbering...\string\endnumbering\space without \string\pstart

\led@warn@BadAction

150 \newcommand*{\led@warn@BadAction}{%
151 \reledmac@warning{Bad action code, value \next@action.}}

\led@warn@DuplicateLabel
\led@warn@AppLabelOutEdtext 152 \newcommand*{\led@warn@DuplicateLabel}[1]{%
\led@warn@RefUndefined 153 \reledmac@warning{Duplicate definition of label `#1' on page \the\pageno.}}
154 \newcommand*{\led@warn@AppLabelOutEdtext}[1]{%
155 \reledmac@warning{\string\applabel\space outside of \string\edtext\space `#1' on page \the\
156 \newcommand*{\led@warn@RefUndefined}[1]{%
157 \reledmac@warning{Reference `#1' on page \the\pageno\space undefined.
158 Using `000'.}}

\led@warn@NoMarginpars

159 \newcommand*{\led@warn@NoMarginpars}{%
160 \reledmac@warning{You can't use \string\marginpar\space in numbered text}}

\led@warn@BadSidenotemargin

161 \newcommand*{\led@warn@BadSidenotemargin}{%
162 \reledmac@warning{Bad \string\sidenotemmargin\space argument}}

\led@warn@NoIndexFile

163 \newcommand*{\led@warn@NoIndexFile}[1]{%
164 \reledmac@warning{Undefined index file #1}}

\led@warn@SeriesStillExist

165 \newcommand{\led@warn@SeriesStillExist}[1]{%
166 \reledmac@warning{Series #1 is still existing !}}%
167 }%

```

```

\led@err@ManySidenotes
\led@err@ManyLeftnotes 168 \newcommand{\led@err@ManySidenotes}{%
\led@err@ManyRightnotes 169 \ifledRcol{%
170 \reledmac@warning{\itemcount@\space sidenotes on line \the\line@numR\space p. \the\page@numR}%
171 \else%
172 \reledmac@warning{\itemcount@\space sidenotes on line \the\line@num\space p. \the\page@num}%
173 \fi%
174 }%
175 \newcommand{\led@err@ManyLeftnotes}{%
176 \ifledRcol{%
177 \reledmac@warning{\itemcount@\space leftnotes on line \the\line@numR\space p. \the\page@numR}%
178 \else%
179 \reledmac@warning{\itemcount@\space leftnotes on line \the\line@num\space p. \the\page@num}%
180 \fi%
181 }%
182 \newcommand{\led@err@ManyRightnotes}{%
183 \ifledRcol{%
184 \reledmac@warning{\itemcount@\space rightnotes on line \the\line@numR\space p. \the\page@numR}%
185 \else%
186 \reledmac@warning{\itemcount@\space rightnotes on line \the\line@num\space p. \the\page@num}%
187 \fi%
188 }%

\led@err@TooManyColumns
\led@err@UnequalColumns 189 \newcommand*\led@err@TooManyColumns{%
\led@err@LowStartColumn 190 \reledmac@error{Too many columns}{\@ehc}}
\led@err@HighEndColumn 191 \newcommand*\led@err@UnequalColumns{%
\led@err@ReverseColumns 192 \reledmac@error{Number of columns is not equal to the number
193 in the previous row (or \protect\ \space forgotten?)}{\@ehc}}
194 \newcommand*\led@err@LowStartColumn{%
195 \reledmac@error{Start column is too low}{\@ehc}}
196 \newcommand*\led@err@HighEndColumn{%
197 \reledmac@error{End column is too high}{\@ehc}}
198 \newcommand*\led@err@ReverseColumns{%
199 \reledmac@error{Start column is greater than end column}{\@ehc}}

err@EdtextWithoutFootnote
200 \newcommand{\led@err@EdtextWithoutFootnote}{%
201 \reledmac@error{edtext without Xfootnote. Check syntaxis.}{\@ehc}%
202 }%

err@FootnoteWithoutEdtext
203 \newcommand{\led@err@FootnoteWithoutEdtext}{%
204 \reledmac@error{Xfootnote without edtext. Check syntax.}{\@ehc}%
205 }%

rrror@ImakeidxAfterEledmac
206 \newcommand{\led@error@ImakeidxAfterEledmac}{%
207 \reledmac@error{Imakeidx must be loaded before eledmac.}{\@ehc}%
208 }%

```

```

\led@error@IndextoolsAfterEledmac

209 \newcommand{\led@error@IndextoolsAfterEledmac}{%
210   \reledmac@error{Indextools must be loaded before eledmac.}\@ehc}%
211 }%

\led@error@fail@patch@@makecol

212 \newcommand{\led@error@fail@patch@@makecol}{%
213   \reledmac@error{Fail to patch \string\@makecol\space command.}\@ehc}%
214 }%

\led@error@fail@patch@@reinserts

215 \newcommand{\led@error@fail@patch@@reinserts}{%
216   \reledmac@error{Fail to patch \string\@reinserts\space command.}\@ehc}%
217 }%

\led@error@fail@patch@@docclearpage

218 \newcommand{\led@error@fail@patch@@docclearpage}{%
219   \reledmac@error{Fail to patch \string\@docclearpage\space command.}\@ehc}%
220 }%

\led@error@fail@patch@@iiiminipage

221 \newcommand{\led@error@fail@patch@@iiiminipage}{%
222   \reledmac@error{Fail to patch \string\@iiiminipage\space command.}\@ehc}%
223 }%

\led@error@fail@patch@endminipage

224 \newcommand{\led@error@fail@patch@endminipage}{%
225   \reledmac@error{Fail to patch \string\endminipage\space command.}\@ehc}%
226 }%

```

## II.7 Gobbling

Here, we define some commands which gobble their arguments.

```

\@gobblethree
\@gobblefour 227 \providecommand*\@gobblethree}[3]{}
\@gobblefive 228 \providecommand*\@gobblefour}[4]{}
229 \providecommand*\@gobblefive}[5]{}

```

## II.8 Miscellaneous commands

`\showlemma` `\showlemma{<lemma>}` typesets the lemma text in the body. It depends on the option.

```

230 \ifledfinal
231   \newcommand*\showlemma}[1]{#1}
232 \else
233   \newcommand*\showlemma}[1]{\underline{#1}}
234 \fi
235

```



`\linenumberlist` The code for the `\linenumberlist` mechanism was given to Peter Wilson by Wayne Sullivan on 2004/02/11.  
Initialize it as `\empty`.  
236 `\let\linenumberlist=\empty`  
237

`\@l@tempcnta` In imitation of  $\TeX$ , we create a couple of scratch counters.  
`\@l@tempcntb`  $\TeX$  already defines `\@tempcnta` and `\@tempcntb` but Peter Wilson found in the past that it can be dangerous to use these (for example one of the AMS packages did something nasty to the `ccaption` package's use of one of these).  
238 `\newcount\@l@tempcnta \newcount\@l@tempcntb`

## II.9 Prepare reledpar

`\ifnumberingR` In preparation for the `reledpar` package, these are related to the 'right' text of parallel texts (when `\ifl@dpairing` is TRUE). They are explained in the `eledpar` manual.  
`\ifl@dpairing`  
`\ifl@dpaging`  
`\l@dpagingtrue` 239 `\newif\ifl@dpairing`  
`\l@dpagingfalse` 240 `\newif\ifl@dpaging%`  
`\ifl@dprintingpages` 241 `\newif\ifl@dprintingpages%`  
`\l@dprintingpagestrue` 242 `\newif\ifl@dprintingcolumns%`  
`\l@dprintingpagesfalse` 243 `\newif\ifpst@rtedL`  
`\ifl@dprintingcolumns` 244 `\newcount\l@dnumpstartsL`  
`\l@dprintingcolumnstrue` `\ifledRcol` is set to true in the Rightside environnement. It must be not confused with `\ifledRcol@` which is set to true when a right line is processed, in `\Pages` or `\Columns`.  
`\l@dprintingcolumnsfalse`  
`\l@dpairingtrue`  
`\l@dpairingfalse` 245 `\newif\ifledRcol`  
`\ifpst@rtedL` 246 `\newif\ifledRcol@`  
`\pst@rtedLtrue` The `\ifnumberingR` flag is set to true if we're within a right text numbered section.  
`\pst@rtedLfalse` 247 `\newif\ifnumberingR`  
`\l@dnumpstartsL`  
`\ifledRcol`  
`\ifledRcol@`

## III Sectioning commands

`\section@num` You use `\beginnumbering` and `\endnumbering` 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.  $\TeX$  will maintain and display a 'section number' as a count named `\section@num` that counts how many `\beginnumbering` and `\resumenummering` commands have appeared; it need not be related to the logical divisions of your text.

`\extensionchars` Each section will read and write an associated 'line-list file', containing information used to do the numbering; the file will be called `\jobname.nn`, where `nn` is the section number. However, you may direct that an extra string be added before the `nn` in that filename, in order to distinguish these temporary files from others: that

string is called `\extensionchars`. Initially it's empty, since different operating systems have greatly varying ideas about what characters are permitted in file names. So `\renewcommand{\extensionchars}{-}` gives temporary files called `jobname.-1`, `jobname.-2`, etc.

```
248 \newcount\section@num
249 \section@num=0
250 \let\extensionchars=\empty
```

`\ifnumbering` The `\ifnumbering` flag is set to true if we are within a numbered section (that is, between `\beginnumbering` and `\endnumbering`). You can use `\ifnumbering` in your own code to check whether you are in a numbered section, but do not change the flag's value.

```
251 \newif\ifnumbering
```

`\beginnumbering` `\beginnumbering` begins a section of numbered text. When it is 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. `\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 is 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 `\l@dnumstartsL` — the number of chunks to be processed.
- set `\ifpst@rtedL` to FALSE.

```
252 \newcommand*{\beginnumbering}{%
253   \ifnumbering
254     \led@err@NumberingStarted
255   \endnumbering
256 \fi
257 \global\numberingtrue
258 \global\advance\section@num \@ne
259 \initnumbering@reg
260 \message{Section \the\section@num }%
261 \line@list@stuff{\jobname.\extensionchars\the\section@num}%
262 \l@dend@stuff
263 \setcounter{pstart}{1}
264 \ifl@dpairing
265   \global\l@dnumstartsL \z@
266   \global\pst@rtedLfalse
```

The tools for section's title commands are called:

- 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.

```

267 \else
268 \begingroup
269 \global\@afterindenttrue%In order to reestablish normal feature if the \begingroup was not here
270 \initnumbering@quote
271 \ifwidthliketwocolumns%
272 \csuse{setwidthliketwocolumns@\columns@position}%
273 \csuse{setpositionliketwocolumns@\columns@position}%
274 \fi%
275 \fi
276 \gdef\eled@sections@{ }%
277 \if@noeled@sec\else%
278 \makeatletter\inputifexists{\jobname.eledsec\the\section@num}{ }\makeatother%
279 \immediate\openout\eled@sectioning@out=\jobname.eledsec\the\section@num\relax%
280 \fi%
281 }
282 \newcommand*{\initnumbering@reg}{%
283 \global\pst@rtedLfalse
284 \global\l@dnumpststartL \z@
285 \global\absline@num \z@
286 \gdef\normal@page@break{ }
287 \gdef\l@prev@pb{ }
288 \gdef\l@prev@nopb{ }
289 \global\line@num \z@
290 \global\subline@num \z@
291 \global\@lock \z@
292 \global\sub@lock \z@
293 \global\sublines@false
294 \global\let\next@page@num=\relax
295 \global\let\sub@change=\relax
296 \resetprevline@
297 \resetprevpage@num
298 }
299

```

`\endnumbering` `\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.

```

300 \def\endnumbering{%
301 \ifnumbering
302 \global\numberingfalse
303 \normal@pars
304 \ifnum\l@dnumpststartL=0%
305 \led@err@NumberingWithoutPstart%
306 \fi%
307 \ifl@dpairing
308 \global\pst@rtedLfalse

```

```

309 \else
310 \ifx\insertlines@list\empty\else
311 \global\noteschanged@true
312 \fi
313 \ifx\line@list\empty\else
314 \global\noteschanged@true
315 \fi
316 \fi
317 \ifnoteschanged@
318 \led@mess@NotesChanged
319 \fi
320 \else
321 \led@err@NumberingNotStarted
322 \fi
323 \autoparfalse
324 \if@noeled@sec\else%
325 \immediate\closeout\eled@sectioning@out%
326 \fi%
327 \ifl@dpairing\else
328 \global\l@dnumstartsl=\z@%
329 \endgroup
330 \fi
331 }

```

`\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.<sup>23</sup>

```

332 \newcommand{\pausenumbering}{%
333 \ifautopar\global\autopar@pausetrue\fi%
334 \endnumbering\global\numberingtrue}

```

The `\resumenumbers` 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 `\resumenumbers` to ensure that `\pausenumbering` was actually invoked.

```

335 \newcommand*{\resumenumbers}{%
336 \ifnumbering
337 \ifautopar@pause\autopar\fi
338 \global\pst@rtedLtrue
339 \global\advance\section@num \@ne
340 \led@mess@SectionContinued{\the\section@num}%
341 \line@list@stuff{\jobname.\extensionchars\the\section@num}%
342 \l@dend@stuff
343 \ifl@dpairing\else%
344 \begingroup%
345 \initnumbering@quote%
346 \ifwidthliketwocolumns%
347 \csuse{setwidthliketwocolumns@\columns@position}%
348 \csuse{setpositionliketwocolumns@\columns@position}%

```

<sup>23</sup>Peter Wilson's thanks to Wayne Sullivan, who suggested the idea behind these macros.

```

349         \fi%
350     \fi%
351 \else
352     \led@err@NumberingShouldHaveStarted
353     \endnumbering
354     \beginnumbering
355 \fi}
356
357

```

## IV List macros

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.

The historical list tools of ledmac are kept, because in many cause there are more useful than etoolbox’s lists. They allows to get and delete the first element of a list in one operation. They also expands the items add to the list.

However, etoolbox’s lists are more useful to loop on them. Consequently, depending of what we need, we use one or either.

It could be nice to unify them to the  $\text{\LaTeX}$ 3 list, however such migration would take quite time with some risk of error, for a gain which will be minor.

`\list@create` The `\list@create` macro creates a new list. This macro does not do anything beyond initializing an empty list macro.

```

358 \newcommand*{\list@create}[1]{%
359     \global\let#1=\empty%
360 }%

```

`\list@clear` The `\list@clear` macro just initializes a list to the empty list; it is no different from `\list@create` in its effect, but it is in its semantic .

```

361 \newcommand*{\list@clear}[1]{%
362     \global\let#1=\empty%
363 }

```

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

```

364 \newtoks\led@toksa \newtoks\led@toksb
365 \global\led@toksa={\}
366 \long\def\xright@appenditem#1\to#2{%
367     \global\led@toksb=\expandafter{#2}%
368     \xdef#2{\the\led@toksb\the\led@toksa\expandafter{#1}}%

```

```

369 \global\led@toksb={}}

\yleft@appenditem \yleft@appenditem expands an item and appends it to the left end of a list macro; it
is otherwise identical to \xright@appenditem.
370 \long\def\yleft@appenditem#1\to#2{%
371 \global\led@toksb=\expandafter{#2}%
372 \xdef#2{\the\led@toksa\expandafter{#1}\the\led@toksb}%
373 \global\led@toksb={}}

\gl@p The \gl@p macro removes the leftmost item from a list and places it in a control se-
quence. You say \gl@p\l\to\z (where \l is the list macro, and \z receives the left
item). \l is assumed nonempty:use \ifx\l\empty to test for an empty \l. The control
sequences created by \gl@p are all global.
374 \def\gl@p#1\to#2{\expandafter\gl@poff#1\gl@poff#1#2}
375 \long\def\gl@poff\#1#2\gl@poff#3#4{\gdef#4{#1}\gdef#3{#2}}
376

```

## V Line counting

### V.1 Choosing the system of lineation

Line number can be reset at each section (default) ; at each page ; at each pstart. Here we define internal codes for these systems and the macros.

```

\ifbypstart@ The \ifbypage@ and \ifbypstart@ flag specifie the current lineation system:
\byppstart@true
\byppstart@false
\ifbypage@
\byppage@true
\byppage@false reledmac will use the line-of-section system unless instructed otherwise.
377 \newif\ifbypage@
378 \newif\ifbypstart@

```

The \ifbypage@R and \ifbypstart@R flag specifie the current lineation for right side in case of using reledpar. They are now defined because they are used in some specific code. reledpar will use the line-of-section system unless instructed otherwise.

```

\ifbypage@R
\ifbypstart@R 379 \newif\ifbypage@R
380 \newif\ifbypstart@R

```

```

\lineation \lineation{<word>} is the macro you use to select the lineation system. Its argument
is a string: either page, section or pstart.
381 \newcommand*{\lineation}[1]{{

```

We can't change the lineation system inside numbering section.

```

382 \ifnumbering
383 \led@err@LineationInNumbered
384 \else
385 % \end{macrocode}
386 % If the argument is \verb+page+.
387 % \begin{macrocode}
388 \def\@tempa{#1}\def\@tempb{page}%
389 \ifx\@tempa\@tempb
390 \global\bypage@true
391 \global\bystart@false
392 \unless\ifnocritical@%
393 \Xpstart[] [false]%
394 \fi%
395 % \end{macrocode}
396 % If the argument is \verb+pstart+.
397 % \begin{macrocode}
398 \else
399 \def\@tempb{pstart}%
400 \ifx\@tempa\@tempb
401 \global\bypage@false
402 \global\bystart@true
403 \unless\ifnocritical@%
404 \Xpstart%
405 \fi%
406 % \end{macrocode}
407 % And finally, if the argument is \verb+section+ (default).
408 % \begin{macrocode}
409 \else
410 \def\@tempb{section}
411 \ifx\@tempa\@tempb
412 \global\bypage@false
413 \global\bystart@false
414 \unless\ifnocritical@%
415 \Xpstart[] [false]%
416 \fi%
417 \else
418 \led@warn@BadLineation
419 \fi
420 \fi
421 \fi
422 \fi}}

```

In other case, it is an error.

## V.2 Line number margin

`\linenummargin` `\linenummargin{<word>}` specify which margin line numbers are in; it takes one argument, a string, which value can be left ; right; inner or outer.

`\line@margin`

`\l@dgetline@margin`

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

```

423 \newcount\line@margin
424
425 \newcommand*\linenummargin}[1]{%
426   \l@getline@margin{#1}%
427   \ifnum\l@ldtempcntb>\m@ne
428     \ifledRcol
429       \global\line@marginR=\l@ldtempcntb
430     \else
431       \global\line@margin=\l@ldtempcntb
432     \fi
433   \fi}}
434
435 \newcommand*\l@getline@margin}[1]{%
436   \def\@tempa{#1}\def\@tempb{left}%
437   \ifx\@tempa\@tempb
438     \l@ldtempcntb \z@
439   \else
440     \def\@tempb{right}%
441     \ifx\@tempa\@tempb
442       \l@ldtempcntb \@ne
443     \else
444       \def\@tempb{outer}%
445       \ifx\@tempa\@tempb
446         \l@ldtempcntb \tw@
447       \else
448         \def\@tempb{inner}%
449         \ifx\@tempa\@tempb
450           \l@ldtempcntb \thr@@
451         \else
452           \led@warn@BadLinenummargin
453           \l@ldtempcntb \m@ne
454         \fi
455       \fi
456     \fi
457   \fi}
458
```

### V.3 Line number initialization and increment

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

```

459 \newcounter{firstlinenum}
460 \setcounter{firstlinenum}{5}

```



```

461 \newcounter{linenumincrement}
462 \setcounter{linenumincrement}{5}

```

`\c@firstsublinenum`    The following parameters are just like `firstlinenum` and `linenumincrement`, but for  
`\c@sublinenumincrement`    sub-line numbers. `sublinenumincrement` must be at least 1.

```

463 \newcounter{firstsublinenum}
464 \setcounter{firstsublinenum}{5}
465 \newcounter{sublinenumincrement}
466 \setcounter{sublinenumincrement}{5}
467

```

`\firstlinenum`    These macros can be used to set the corresponding counters.

```

\linenumincrement 468
\firstsublinenum 469 \newcommand*{\firstlinenum}[1]{%
\sublinenumincrement 470 \ifledRcol%
471 \setcounter{firstlinenumR}{#1}%
472 \else%
473 \setcounter{firstlinenum}{#1}%
474 \fi%
475 }
476 \newcommand*{\linenumincrement}[1]{%
477 \ifledRcol%
478 \setcounter{linenumincrementR}{#1}%
479 \else%
480 \setcounter{linenumincrement}{#1}%
481 \fi%
482 }
483 \newcommand*{\firstsublinenum}[1]{%
484 \ifledRcol%
485 \setcounter{firstsublinenumR}{#1}%
486 \else%
487 \setcounter{firstsublinenum}{#1}%
488 \fi%
489 }
490 \newcommand*{\sublinenumincrement}[1]{%
491 \ifledRcol%
492 \setcounter{sublinenumincrementR}{#1}%
493 \else%
494 \setcounter{sublinenumincrement}{#1}%
495 \fi%
496 }
497

```

## V.4 Line number locking

`\lockdisp`    When line locking is being used, the `\lockdisp{⟨word⟩}` macro specifies whether a  
`\lock@disp`    line number—if one is due to appear—should be printed on the first printed line or on  
`\l@getlock@disp`    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.

```

498 \newcount\lock@disp
499 \newcommand{\lockdisp}[1]{%
500   \l@getlock@disp{#1}%
501   \ifnum\@l@tempcntb>\m@ne
502     \global\lock@disp=\@l@tempcntb
503   \else
504     \led@warn@BadLockdisp
505   \fi}}
506 \newcommand*{\l@getlock@disp}[1]{
507   \def\@tempa{#1}\def\@tempb{first}%
508   \ifx\@tempa\@tempb
509     \@l@tempcntb \z@
510   \else
511     \def\@tempb{last}%
512     \ifx\@tempa\@tempb
513       \@l@tempcntb \@ne
514     \else
515       \def\@tempb{all}%
516       \ifx\@tempa\@tempb
517         \@l@tempcntb \tw@
518       \else
519         \@l@tempcntb \m@ne
520       \fi
521     \fi
522   \fi}
523

```

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

```

524 \newcount\sublock@disp
525 \newcommand{\sublockdisp}[1]{%
526   \l@getlock@disp{#1}%
527   \ifnum\@l@tempcntb>\m@ne
528     \global\sublock@disp=\@l@tempcntb
529   \else
530     \led@warn@BadSublockdisp
531   \fi}}
532

```

## V.5 Line number style

\linenumberstyle We provide a mechanism for using different representations of the line numbers, not  
\linenumrep just the normal arabic.  
\linenumr@p NOTE: In v0.7 \linenumrep and \sublinenumrep replaced the internal \linenumr@p  
\sublinenumberstyle and \sublinenumr@p.  
\sublinenumrep \linenumberstyle and \sublinenumberstyle are user level macros for setting  
\sublinenumr@p the number representation (\linenumrep and \sublinenumrep) for line and sub-line  
numbers.

```

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

```

Initialise the number styles to arabic.

```

537 \linenumberstyle{arabic}
538 \let\linenumr@p\linenumrep
539 \sublinenumberstyle{arabic}
540 \let\sublinenumr@p\sublinenumrep
541

```

## V.6 Line number printing

`\leftlinenum` and `\rightlinenum` are the macros that are called to print marginal line numbers on a page, for left- and right-hand margins respectively. They are made easy to access and change, since you may want to change the styling in some way. These standard versions illustrate the general sort of thing that will be needed; they are 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 will 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 subtitle) number.

The original `\numlabfont` specification is equivalent to the  $\TeX$  `\scriptsize` for a 10pt document.

```

542 \newlength{\linenumsep}
543 \setlength{\linenumsep}{1pc}
544 \newcommand*{\numlabfont}{\normalfont\scriptsize}
545 \newcommand*{\ledlinenum}{%
546   \bgroup%
547   \ifluatex%
548     \luatextextdir TLT%
549   \fi%
550   \numlabfont\linenumrep{\line@num}%
551   \ifsublines@
552     \ifnum\subline@num>0\relax
553       \unskip\fullstop\sublinenumrep{\subline@num}%
554     \fi
555   \fi%
556   \egroup%
557 }%
558
559 \newcommand*{\leftlinenum}{%
560   \ledlinenum
561   \kern\linenumsep}

```

```

562 \newcommand*{\rightlinenum}{%
563   \kern\linenumsep
564   \ledlinenum}
565

```

## V.7 Line number counters and lists

Footnote references using line numbers rather than symbols can't be generated in one pass, because we do not 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  $\text{\LaTeX}$  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.

<pre> \line@num 566 \newcount\line@num </pre>	<p>The count <code>\line@num</code> stores the line number that is used in marginal line numbering and in notes: counting either by section, page or pstart, depending on your choice for this section. This may be qualified by <code>\subline@num</code>.</p>
<pre> \subline@num 567 \newcount\subline@num </pre>	<p>The count <code>\subline@num</code> stores a sub-line number that qualifies <code>\line@num</code>. 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.</p>
<pre> \ifsublines@ \sublines@true \sublines@false </pre>	<p>We maintain an associated flag, <code>\ifsublines@</code>, to tell us whether we're within a sub-line range or not.</p> <p>You may wonder why we do not just use the value of <code>\subline@num</code> 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.</p>
<pre> 568 \newif\ifsublines@ </pre>	
<pre> \absline@num </pre>	<p>The count <code>\absline@num</code> stores the absolute number of lines since the start of the section: that is, the number we have actually printed, no matter what numbers we attached to them. This value is never printed on an output page, though <code>\line@num</code> will often be equal to it. It is used internally to keep track of where notes are to appear and where</p>

new pages start: using this value rather than `\line@num` is a lot simpler, because it does not depend on the lineation system in use.

569 `\newcount\absline@num`

We will call `\absline@num` numbers “absolute” numbers, and `\line@num` and `\subline@num` numbers “visible” numbers.

## V.8 Line number locking counter

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

570 `\newcount\@lock`

571 `\newcount\sub@lock`

## V.9 Line number associated to lemma

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

`\insertlines@list`  
`\actionlines@list`  
`\actions@list`

- `\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/cm/r/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 `reledmac` what action it is supposed to take at each of these lines. One action, the page-start action, is generated behind the scenes by `reledmac` 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  $\geq 0$ ) 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 `reledmac` computes the visible line numbers from the absolute line numbers with reference to the other action codes and the settings they invoke; it does not require an entry in the action-code list for every line.

Here are the commands to create these lists:

```
572 \list@create{\line@list}
573 \list@create{\insertlines@list}
574 \list@create{\actionlines@list}
575 \list@create{\actions@list}
576
```

```
\page@num We will need some counts while we read the line-list, for the page number and the ending
\endpage@num page, line, and sub-line numbers. Some of these will be used again later on, when we
\endline@num are acting on the data in our list macros.
\endsubline@num 577 \newcount\page@num
578 \newcount\endpage@num
579 \newcount\endline@num
580 \newcount\endsubline@num
```

```
\ifnoteschanged@ If the number of the footnotes in a section is different from what it was during the last
\noteschanged@true run, or if this is the very first time you've run  $\LaTeX$ , on this file, the information from
\noteschanged@false 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 do
not 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 is no fix needed to the document.
The \ifnoteschanged@ flag is set if such a change in the number of notes is discovered
at any point.
```

```
581 \newif\ifnoteschanged@
```

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

```
\resetprevline@
582 \newcommand*{\resetprevline@}{%
583 \def\do##1{\global\csundef{prevline##1}}%
584 \dolistloop{@series}%
585 }
```

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

```
\resetprevpage@
586 \newcommand*{\resetprevpage@num}{%
587   \def\do##1{\ifcsdef{prevpage##1@num}{\global\csname prevpage##1@num\endcsname=0}{}}%
588   \dolistloop{\@series}%
589 }
```

## V.10 Reading the line-list file

`\read@linelist` `\read@linelist{<file>}` is the control sequence that is called by `\beginnumbering` (via `\line@list@stuff`) to open and process a line-list file; its argument is the name of the file. First, it clear all previous line's list.

```
590 \newread\@inputcheck
591 \newcommand*{\read@linelist}[1]{%
592   \ifledRcol%
593     \list@clearing@regR%
594   \else%
595     \list@clearing@reg%
596   \fi%
```

When using `reledpar`, make sure that the `\maxlinesinpar@list` is empty (otherwise things will be thrown out of kilter if there is any old stuff still hanging in there).

```
597 \list@clear{\maxlinesinpar@list}
```

Now get the file and interpret it. When the file is there we start a new group and make some special definitions we will need to process it. It is a sequence of  $\TeX$  commands, but they require a few special settings. We make `[` and `]` become grouping characters: they are 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 is easier to just use something other than real braces. `@` must become a letter, since this is run in the ordinary  $\LaTeX$  context. We ignore carriage returns, since if we are 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 `\resumenummering`, those things should still have the values they had when `\pausenummering` was executed.

If the file is not there, we print an informative message.

Now, after these preliminaries, we start interpreting the file.

```
598 \get@linelistfile{#1}%
599 \endgroup
```

When the reading is done, we are 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.

```

600 \ifledRcol
601   \global\page@numR=\m@ne
602   \ifx\actionlines@listR\empty
603     \gdef\next@actionlineR{1000000}%
604   \else
605     \gl@p\actionlines@listR\to\next@actionlineR
606     \gl@p\actions@listR\to\next@actionR
607   \fi
608 \else
609   \global\page@num=\m@ne
610   \ifx\actionlines@list\empty
611     \gdef\next@actionline{1000000}%
612   \else
613     \gl@p\actionlines@list\to\next@actionline
614     \gl@p\actions@list\to\next@action
615   \fi
616 \fi
617 }

```

`\list@clearing@reg` Clears the lists for `\read@linelist`

```

618 \newcommand*{\list@clearing@reg}{%
619   \list@clear{\line@list}%
620   \list@clear{\insertlines@list}%
621   \list@clear{\actionlines@list}%
622   \list@clear{\actions@list}%
623   \list@clear{\linesinpar@listL}%
624   \list@clear{\linesonpage@listL}%
625 }%

```

`\get@linelistfile` reledmac can take advantage of the  $\TeX$  ‘safe file input’ macros to get the line-list file.

```

626 \newcommand*{\get@linelistfile}[1]{%
627   \InputIfFileExists{#1}{%
628     \global\noteschanged@false
629     \begingroup
630       \catcode`\[=1 \catcode`\]=2
631       \makeatletter \catcode`\^M=9}{%
632     \led@warn@NoLineFile{#1}%
633     \global\noteschanged@true
634     \begingroup}%
635 }
636

```

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 would have to do some file renaming outside of  $\TeX$  for that to work. We have retained this slower approach to avoid that sort of hacking about, but have provided the `\pausenumbering` and `\resumenumbering` macros to help you if you run into macro memory limitations (see 4.2.7 p. 17 above).

## V.11 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 deliberately 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 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`.

`\line@list@version` The `\line@list@version` check if the line-list file does not refers to the older commands of `reledmac`. In this case, we stop reading the line-list file. Consequently, `\line@list@version` must be the first line of a line-number file.

```

637 \newcommand{\line@list@version}[1]{%
638   \IfStrEq{#1}{\this@line@list@version}%
639   }{%
640     {\ifledRcol%
641       \led@warn@Obsolete{\jobname.\extensionchars\the\section@num}%
642       \else%
643       \led@warn@Obsolete{\jobname.\extensionchars\the\section@num}%
644       \fi%
645       \endinput%
646     }%
647 }%
```

`\@nl` `\@nl` does everything related to the start of a new line of numbered text.

`\@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 *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}
```

We do not (yet) use the printed number (i.e., the `\thepage`) but it may come in handy later. The macro `\fix@page` checks if a new page has started.

Exactly what `\@nl` does depends on whether right text is being processed. That's why many code is defined in `\@nl@reg` or `\nl@regR`.

```

648
649 \newcommand*{\@nl}[2]{%
650   \fix@page{#1}%
651   \ifledRcol%
652     \@nl@regR%
653   \else%
654     \@nl@reg%
655   \fi%
656 }
657 \newcommand*{\@nl@reg}{%
658   \ifx\l@dchset@num\relax \else
659     \advance\absline@num \@ne
660     \set@line@action
661     \let\l@dchset@num=\relax
662     \advance\absline@num \m@ne
663     \advance\line@num \m@ne
664   \fi

```

First increment the absolute line-number, and perform deferred actions relating to page starts and sub-lines.

```

665   \advance\absline@num \@ne
666     \ifx\next@page@num\relax \else
667       \page@action
668       \let\next@page@num=\relax
669     \fi
670     \ifx\sub@change\relax \else
671       \ifnum\sub@change>\z@
672         \sublines@true
673       \else
674         \sublines@false
675       \fi
676       \sub@action
677       \let\sub@change=\relax
678     \fi

```

Fix the lock counters, if necessary. A value of 1 is advanced to 2; 3 advances to 0; other values are unchanged.

```

679     \ifcase\@lock
680       \or
681         \@lock \tw@
682       \or \or
683         \@lock \z@
684     \fi
685     \ifcase\sub@lock
686       \or
687         \sub@lock \tw@

```

```

688         \or \or
689         \sub@lock \z@
690     \fi
    Now advance the visible line number, unless it has been locked.
691     \ifsublines@
692         \ifnum\sub@lock<\tw@
693             \advance\subline@num \@ne
694         \fi
695     \else
696         \ifnum\@lock<\tw@
697             \advance\line@num \@ne \subline@num \z@
698         \fi
699     \fi}
700

```

`\last@page@num` `\fix@page` basically replaces `\@page`. It determines whether or not a new page has been started, based on the page values held by `\@n1`.

```

701 \newcount\last@page@num
702 \last@page@num=-10000
703
704 \newcommand*{\fix@page}[1]{%
705     \ifledRcol
706         \ifnum #1=\last@page@numR
707         \else
708             \ifbypage@R
709                 \line@numR \z@ \subline@numR \z@
710             \fi
711             \page@numR=#1\relax
712             \last@page@numR=#1\relax
713             \def\next@page@numR{#1}%
714         \fi
715     \else
716         \ifnum #1=\last@page@num
717         \else
718             \ifbypage@
719                 \line@num \z@ \subline@num \z@
720             \fi
721             \page@num=#1\relax
722             \last@page@num=#1\relax
723             \def\next@page@num{#1}%
724             \listxadd{\normal@page@break}{\the\absline@num}
725         \fi
726     \fi}

```

`\@pend` These do not do anything at this point, but will have been added to the auxiliary file(s) if the `reledpar` package has been used. They are just here to stop `reledmac` from moaning

`\@pendR` if the `reledpar` is used for one run and then not for the following one.

`\@lopL`

`\@lopR`

```

727 \newcommand*{\@pend}[1]{%
728 \newcommand*{\@pendR}[1]{%

```

```

729 \newcommand*{\@lopL}[1]{%
730 \newcommand*{\@lopR}[1]{%
731

```

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

```

732 \newcommand*{\sub@on}{\ifsublines@
733     \let\sub@change=\relax
734 \else
735     \def\sub@change{1}%
736 \fi}
737 \newcommand*{\sub@off}{\ifsublines@
738     \def\sub@change{-1}%
739 \else
740     \let\sub@change=\relax
741 \fi}
742

```

`\@adv` The `\@adv{<num>}` macro advances the current visible line number by the amount specified as its argument. This is used to implement `\advanceline`.

```

743
744 \newcommand*{\@adv}[1]{%
745     \ifsublines@
746         \ifledRcol
747             \advance\subline@numR by #1\relax
748             \ifnum\subline@numR<\z@
749                 \led@warn@BadAdvancelineSubline
750                 \subline@numR \z@
751             \fi
752         \else
753             \advance\subline@num by #1\relax
754             \ifnum\subline@num<\z@
755                 \led@warn@BadAdvancelineSubline
756                 \subline@num \z@
757             \fi
758         \fi
759     \else
760         \ifledRcol
761             \advance\line@numR by #1\relax
762             \ifnum\line@numR<\z@
763                 \led@warn@BadAdvancelineLine
764                 \line@numR \z@
765             \fi
766         \else
767             \advance\line@num by #1\relax
768             \ifnum\line@num<\z@
769                 \led@warn@BadAdvancelineLine
770                 \line@num \z@

```

```

771     \fi
772     \fi
773     \fi
774     \set@line@action}
775

```

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

```

776
777 \newcommand*{\@set}[1]{%
778   \ifledRcol
779     \ifsublines@
780       \subline@numR=#1\relax
781     \else
782       \line@numR=#1\relax
783     \fi
784     \set@line@action
785   \else
786     \ifsublines@
787       \subline@num=#1\relax
788     \else
789       \line@num=#1\relax
790     \fi
791     \set@line@action
792   \fi}
793

```

`\l@d@set` The `\l@d@set{<num>}` macro sets the line number for the next `\pstart` to the value specified as its argument. This is used to implement `\setlinenum`.

`\l@dchset@num` `\l@dchset@num` is a flag to the `\@nl?` macro. If it is not `\relax` then a linenum change is to be done.

```

794
795 \newcommand*{\l@d@set}[1]{%
796   \ifledRcol
797     \line@numR=#1\relax
798     \advance\line@numR \@ne
799     \def\l@dchset@num{#1}
800   \else
801     \line@num=#1\relax
802     \advance\line@num \@ne
803     \def\l@dchset@num{#1}
804   \fi}
805 \let\l@dchset@num\relax
806

```

`\page@action` `\page@action` adds an entry to the action-code list to change the page number.

```

807
808 \newcommand*{\page@action}{%
809   \ifledRcol

```

```

810 \xright@appenditem{\the\absline@numR}\to\actionlines@listR
811 \xright@appenditem{\next@page@numR}\to\actions@listR
812 \else
813 \xright@appenditem{\the\absline@num}\to\actionlines@list
814 \xright@appenditem{\next@page@num}\to\actions@list
815 \fi}

```

`\set@line@action` `\set@line@action` adds an entry to the action-code list to change the visible line number.

```

816
817 \newcommand*{\set@line@action}{%
818 \ifledRcol
819 \xright@appenditem{\the\absline@numR}\to\actionlines@listR
820 \ifsublines@
821 \@l@dttempcnta=-\subline@numR
822 \else
823 \@l@dttempcnta=-\line@numR
824 \fi
825 \advance\@l@dttempcnta by -5000\relax
826 \xright@appenditem{\the\@l@dttempcnta}\to\actions@listR
827 \else
828 \xright@appenditem{\the\absline@num}\to\actionlines@list
829 \ifsublines@
830 \@l@dttempcnta=-\subline@num
831 \else
832 \@l@dttempcnta=-\line@num
833 \fi
834 \advance\@l@dttempcnta by -5000\relax
835 \xright@appenditem{\the\@l@dttempcnta}\to\actions@list
836 \fi}

```

`\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 `\ifsublines@` flag.

```

837
838 \newcommand*{\sub@action}{%
839 \ifledRcol
840 \xright@appenditem{\the\absline@numR}\to\actionlines@listR
841 \ifsublines@
842 \xright@appenditem{-1001}\to\actions@listR
843 \else
844 \xright@appenditem{-1002}\to\actions@listR
845 \fi
846 \else
847 \xright@appenditem{\the\absline@num}\to\actionlines@list
848 \ifsublines@
849 \xright@appenditem{-1001}\to\actions@list
850 \else
851 \xright@appenditem{-1002}\to\actions@list
852 \fi
853 \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 is 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.

```

854 \newcommand*{\lock@on}{\futurelet\next\do@lockon}
855
856 \newcommand*{\do@lockon}{%
857   \ifx\next\lock@off
858     \global\let\lock@off=\skip@lockoff
859   \else
860     \ifledRcol
861       \do@lockonR
862     \else
863       \do@lockonL
864     \fi
865   \fi}
866
867
868 \newcommand*{\do@lockonL}{%
869   \xright@appenditem{\the\absline@num}\to\actionlines@list
870   \ifsublines@
871     \xright@appenditem{-1005}\to\actions@list
872     \ifnum\sub@lock=\z@
873       \sub@lock \@ne
874     \else
875       \ifnum\sub@lock=\thr@@
876         \sub@lock \@ne
877       \fi
878     \fi
879   \else
880     \xright@appenditem{-1003}\to\actions@list
881     \ifnum\@lock=\z@
882       \@lock \@ne
883     \else
884       \ifnum\@lock=\thr@@
885         \@lock \@ne
886       \fi
887     \fi
888   \fi}
889

```

`\lock@off` `\lock@off` adds an entry to the action-code list to turn line number locking off.

```

\do@lockoff 890 \newcommand*{\do@lockoffL}{%
\do@lockoffL 891   \xright@appenditem{\the\absline@num}\to\actionlines@list
\skip@lockoff 892   \ifsublines@
893     \xright@appenditem{-1006}\to\actions@list

```



```

894 \ifnum\sub@lock=\tw@
895 \sub@lock \thr@@
896 \else
897 \sub@lock \z@
898 \fi
899 \else
900 \xright@appenditem{-1004}\to\actions@list
901 \ifnum\@lock=\tw@
902 \@lock \thr@@
903 \else
904 \@lock \z@
905 \fi
906 \fi}
907
908 \newcommand*{\do@lockoff}{%
909 \ifledRcol
910 \do@lockoffR
911 \else
912 \do@lockoffL
913 \fi}
914 \newcommand*{\skip@lockoff}{\global\let\lock@off=\do@lockoff}
915 \global\let\lock@off=\do@lockoff
916

```

`\n@num` These macros implement the `\skipnumbering` command. They use action code 1007.

```

917 \newcommand*{\n@num}{%
918 \ifledRcol%
919 \xright@appenditem{\the\absline@numR}\to\actionlines@listR
920 \xright@appenditem{-1007}\to\actions@listR
921 \else%
922 \xright@appenditem{\the\absline@num}\to\actionlines@list%
923 \xright@appenditem{-1007}\to\actions@list%
924 \fi%
925 }%
926

```

`\n@num@stanza` This macro implements the `\skipnumbering` for stanza command. It uses action code 1008.

```

927 \newcommand*{\n@num@stanza}{%
928 \ifledRcol%
929 \xright@appenditem{\the\absline@numR}\to\actionlines@listR%
930 \xright@appenditem{-1008}\to\actions@listR%
931 \else%
932 \xright@appenditem{\the\absline@num}\to\actionlines@list%%
933 \xright@appenditem{-1008}\to\actions@list%
934 \fi%
935 }

```

`\ifl@dhidenumber` `\hidenumbering` hides number in margin. It uses action code 1009.

`\h@num`

```

936 \newif\ifl@dhidenumber
937 \newcommand*{\hidenumbering}{%
938   \ifl@dhidenumber%
939     \write\linenum@outR{\string\hide@num}%
940   \else%
941     \write\linenum@out{\string\hide@num}%
942   \fi%
943 }%
944 \newcommand*{\hide@num}{%
945   \ifl@dhidenumber%
946     \xright@appenditem{\the\absline@numR}\to\actionlines@listR%
947     \xright@appenditem{-1009}\to\actions@listR%
948   \else%
949     \xright@appenditem{\the\absline@num}\to\actionlines@listR%
950     \xright@appenditem{-1009}\to\actions@listR%
951   \fi%
952 }
```

`\@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`.

```

953   \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`.)

`\dummy@ref` When nesting of `\@ref` commands does occur, it is necessary to temporarily redefine `\@ref` within `\@ref`, so that we are only doing one of these at a time.

```

954 \newcommand*{\dummy@ref}[2]{#2}
```

`\@ref@reg` The first thing `\@ref` (i.e. `\@ref@reg`) itself does is to add the specified number of items to the `\insertlines@list` list.

```

955 \newcommand*{\@ref}[2]{%
956   \ifl@dhidenumber%
957     \@ref@reg{#1}{#2}%
958   \else%
959     \@ref@reg{#1}{#2}%
960   \fi%
961 }%
962 \newcommand*{\@ref@reg}[2]{%
963   \global\insert@count=#1\relax
964   \global\advance\@edtext@level by 1%
965   \loop\ifnum\insert@count>\z@
966     \xright@appenditem{\the\absline@num}\to\insertlines@list
```

```

967 \global\advance\insert@count \m@ne
968 \repeat

```

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.

```

969 \begingroup
970 \let\@ref=\dummy@ref
971 \let\@lopL\@gobble
972 \let\page@action=\relax
973 \let\sub@action=\relax
974 \let\set@line@action=\relax
975 \let\@lab=\relax
976 \let\@lemma=\relax%
977 \let\@sw\@gobblethree%
978 #2
979 \global\endpage@num=\page@num
980 \global\endline@num=\line@num
981 \global\endsubline@num=\subline@num
982 \endgroup

```

Now store all the information about the location of the lemma's start and end in \line@list.

```

983 \xright@appenditem%
984 {\the\page@num|\the\line@num|%
985 \ifsublines@ \the\subline@num \else 0\fi|%
986 \the\endpage@num|\the\endline@num|%
987 \ifsublines@ \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.

```

988 \expandafter\list@create\expandafter{\csname sw@list@edtext@tmp@\the\@edtext@level\endcsname}%
989 \providebool{lemmacommand@\the\@edtext@level}%
990 \boolfalse{lemmacommand@\the\@edtext@level}%

```

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

```

991 #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.

```

992 \ifnum\@edtext@level>0%
993 \def\create@this@edtext@level{\expandafter\list@create\expandafter{\csname sw@list@edtext@\the\@edtext@level\endcsname}%
994 \ifcsundef{sw@list@edtext@\the\@edtext@level}{\create@this@edtext@level}{}%
995 \letcs{\@tmp}{sw@list@edtext@\the\@edtext@level}%
996 \letcs{\@tmpp}{sw@list@edtext@tmp@\the\@edtext@level}%
997 \xright@appenditem{\expandonce\@tmpp}\to\@tmp%
998 \global\cslet{sw@list@edtext@\the\@edtext@level}{\@tmp}%
999 \fi%

```

Decrease edtext level counter.

```
1000 \global\advance\@edtext@level by -1%
1001 }
1002
```

## V.12 Writing to the line-list file

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

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

```
1003 \newwrite\linenum@out
```

`\iffirst@linenum@out@` Once any file is opened on this stream, we keep it open forever, or else switch to another  
`\first@linenum@out@true` file that we keep open. The reason is that we want the output routine to write the page  
`\first@linenum@out@false` number for every page to this file; otherwise we would have to write it at the start of every line. But it is not very easy for the output routine to tell whether an output stream is open or not. There is 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 is set to be true before any `\linenum@out` file is opened. When such a file is opened for the first time, it is 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.

```
1004 \newif\iffirst@linenum@out@
1005 \first@linenum@out@true
```

`\this@line@list@version` The commands allowed in the line-list file and their arguments can change between two version of reledmac. 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 inside are deprecated, and we can't use them.

```
1006 \newcommand{\this@line@list@version}{2}%
```

`\line@list@stuff` 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.

```
1007 \newcommand*{\line@list@stuff}[1]{%
```

First, use the commands of the previous section to interpret the line-list file from the last run.

```
1008 \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.

```

1009 \iffirst@linenum@out@
1010     \immediate\closeout\linenum@out%
1011     \global\first@linenum@out@false%
1012     \immediate\openout\linenum@out=#1\relax%
1013     \immediate\write\linenum@out{\string\line@list@version{\this@line@list@version}}%
1014 \else

```

If we get here, then this is not the first line-list we have seen, so we do not open or close the files immediately.

```

1015     \ifminipage%
1016         \leavevmode%
1017     \fi%
1018     \closeout\linenum@out%
1019     \openout\linenum@out=#1\relax%
1020 \fi}
1021

```

`\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.

```

1022 \newcommand*{\new@line}{%
1023   \IfStrEq{\led@pb@setting}{after}%
1024     {\xifinlist{\the\absline@num}{\l@prev@nopb}%
1025       {\xifinlist{\the\absline@num}{\normal@page@break}%
1026         {\numgdef{\@next@page}{\c@page+1}%
1027           \write\linenum@out{\string\@nl[\@next@page][\@next@page}}%
1028         }%
1029       {\write\linenum@out{\string\@nl[\the\c@page][\thepage}}}%
1030     }%
1031     {\write\linenum@out{\string\@nl[\the\c@page][\thepage}}}%
1032   }%
1033   \IfStrEq{\led@pb@setting}{before}%
1034     {\numdef{\next@absline}{\the\absline@num+1}%
1035     \xifinlist{\next@absline}{\l@prev@nopb}%
1036       {\xifinlist{\the\absline@num}{\normal@page@break}%
1037         {\numgdef{\nc@page}{\c@page+1}%
1038           \write\linenum@out{\string\@nl[\nc@page][\nc@page}}%
1039         }%
1040       {\write\linenum@out{\string\@nl[\the\c@page][\thepage}}}%
1041     }%
1042     {\write\linenum@out{\string\@nl[\the\c@page][\thepage}}}%
1043   }%
1044   }%
1045   \IfStrEqCase{\led@pb@setting}{%
1046     {before}{\relax}{after}{\relax}}{\write\linenum@out{\string\@nl[\the\c@page][\thepage}}}%
1047 }

```

`\if@noneed@Footnote` `\if@noneed@Footnote` is a boolean to check if we have to print a error message when a `\edtext` is called without any critical notes.

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

```

1048 \newif\if@noneed@Footnote%
1049
1050 \newcommand*{\flag@start}{%
1051   \ifledRcol%
1052     \edef\next{\write\linenum@outR{%
1053       \string\@ref[\the\insert@countR] []}%
1054     \next%
1055     \ifnum\insert@countR<1%
1056       \if@noneed@Footnote\else%
1057         \led@err@EdtextWithoutFootnote%
1058       \fi%
1059     \fi%
1060   \else%
1061     \edef\next{\write\linenum@out{%
1062       \string\@ref[\the\insert@count] []}%
1063     \next%
1064     \ifnum\insert@count<1%
1065       \if@noneed@Footnote\else%
1066         \led@err@EdtextWithoutFootnote%
1067       \fi%
1068     \fi%
1069   \fi}%
1070

```

`\startsub` `\startsub` and `\endsub` turn sub-lineation on and off, by writing appropriate instruc-  
`\endsub` tions 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 does not 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 at-  
tached 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.

```

1071
1072
1073 \newcommand*{\startsub}{\dimen0\lastskip
1074   \ifdim\dimen0>0pt \unskip \fi
1075   \ifledRcol \write\linenum@outR{\string\sub@on}%
1076   \else      \write\linenum@out{\string\sub@on}%
1077   \fi
1078   \ifdim\dimen0>0pt \hskip\dimen0 \fi}
1079 \def\endsub{\dimen0\lastskip

```

```

1080 \ifdim\dimen0>0pt \unskip \fi
1081 \ifledRcol \write\linenum@outR{\string\sub@off}%
1082 \else      \write\linenum@out{\string\sub@off}%
1083 \fi
1084 \ifdim\dimen0>0pt \hskip\dimen0 \fi}
1085

```

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

```

1086 \newcommand*{\advanceline}[1]{%
1087 \ifledRcol \write\linenum@outR{\string\@adv[#1]}%
1088 \else      \write\linenum@out{\string\@adv[#1]}%
1089 \fi%
1090 }

```

`\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.

```

1091
1092 \newcommand*{\setline}[1]{%
1093 \ifnum#1<\z@
1094 \led@warn@BadSetline
1095 \else
1096 \ifledRcol \write\linenum@outR{\string\@set[#1]}%
1097 \else      \write\linenum@out{\string\@set[#1]}%
1098 \fi
1099 \fi}
1100

```

`\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.

```

1101
1102 \newcommand*{\setlinenum}[1]{%
1103 \ifnum#1<\z@
1104 \led@warn@BadSetlinenum
1105 \else
1106 \ifledRcol \write\linenum@outR{\string\l@d@set[#1]}
1107 \else      \write\linenum@out{\string\l@d@set[#1]} \fi
1108 \fi}
1109

```

`\startlock` 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.

```

1110
1111 \newcommand*{\startlock}{%
1112 \ifledRcol \write\linenum@outR{\string\lock@on}%
1113 \else      \write\linenum@out{\string\lock@on}%
1114 \fi}
1115 \def\endlock{%

```

```

1116 \ifl@dskipnumber \write\linenum@outR{\string\lock@off}%
1117 \else \write\linenum@out{\string\lock@off}%
1118 \fi}

```

```

\ifl@dskipnumber In numbered text \skipnumbering will suspend the numbering for that particular line.
\ifl@dskipversenumber
\l@dskipnumbertrue 1119 \newif\ifl@dskipnumber
\l@dskipnumberfalse 1120 \newif\ifl@dskipversenumber%
\skipnumbering 1121 \newcommand*{\skipnumbering}{%
1122 \leavevmode%
1123 \ifl@dskipnumber%
1124 \ifinstanza%
1125 \write\linenum@outR{\string\n@num@stanza}%
1126 \else%
1127 \write\linenum@outR{\string\n@num}%
1128 \fi%
1129 \advanceline{-1}%
1130 \else%
1131 \ifinstanza%
1132 \write\linenum@out{\string\n@num@stanza}%
1133 \else%
1134 \write\linenum@out{\string\n@num}%
1135 \fi%
1136 \advanceline{-1}%
1137 \fi%
1138 }%
1139

```

## VI Marking text for notes

The `\edtext` 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.

The `\edtext` macro takes two arguments.

```
\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.

The `\edtext` macro may be used (somewhat) recursively; that is, `\edtext` 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 is quite likely that we will 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 are not 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 `\edtext` will fail if you try to use a copy that is called something other than `\edtext`. In order to handle recursion, `\edtext` needs to redefine its own definition temporarily at one point, and that does not work if the macro you are calling is not actually named `\edtext`. There is no problem as long as `\edtext` is not invoked in the first argument. If you want to call `\edtext` something else, it is best to create instead a macro that expands to an invocation of `\edtext`, rather than copying `\edtext` 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`, VII.2.1 p. 114). 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 do not provide previous-note information, although it is often wanted; your own macros must handle that. We can not do it correctly without keeping track of what kind of notes have gone past: it is not just a matter of remembering the line numbers associated with the previous invocation of `\edtext`, 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.

## VI.1 `\edtext` 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.

`\end@lemmas` To accomodate 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.

```
1140 \list@create{\end@lemmas}
```

`\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 is because nested `\edtexts` macros create nested `\@ref` entries in the line-list file.

```
1141 \newcommand{\dummy@edtext}[2]{#1}
```

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

```
1142 \newcommand{\dummy@edtext@showlemma}[2]{\showlemma{#1}}%
```

We are going to need another macro that takes one argument and ignores it entirely. This is supplied by the  $\text{\TeX}$  `\gobble{<arg>}`.

`\no@expands` We need to turn off macro expansion for certain sorts of macros we are 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.<sup>24</sup> 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 is 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— $\text{\TeX}$  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  $\text{\TeX}$`  has this sort of problem as well, but is not 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 `reledmac` 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 `reledmac` 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.

---

<sup>24</sup>Since 'control sequences equivalent to characters are not expandable'—*The  $\text{\TeX}$ book*, answer to Exercise 20.14.

(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. A simpler solution is to avoid active character, using Lua<sub>TeX</sub> or Xe<sub>La</sub>TeX.)

```

1143 \newcommand*\no@expands{%
1144   \let\select@lemmafnt=0%
1145   \let\startsub=\relax \let\endsub=\relax
1146   \let\startlock=\relax \let\endlock=\relax
1147   \let\edlabel=\@gobble
1148   \let\setline=\@gobble \let\advanceline=\@gobble
1149   \let\sameword\sameword@inedtext%
1150   \let\edtext=\dummy@edtext
1151   \l@dtabnoexpands
1152   \morenoexpands}
1153 \let\morenoexpands=\relax
1154

```

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

```

1155 \newcommand{\@tag}{}

```

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

```

1156 \newcount\@edtext@level%
1157 \@edtext@level=0%

```

`\edtext` When executed, `\edtext` first ensures that we are in horizontal mode.

```

1158 \newcommand{\edtext}[2]{\leavevmode%

```

Then, check if we are in a numbered paragraph (`\pstart... \pend`).

```

1159   \ifnumberedpar%

```

We increase the `\@edtext@` counter to know in which level of `\edtext` we are.

```

1160     \global\advance\@edtext@level by 1%

```

By default, we do not use `\lemma`

```

1161     \global\@lemmacommand@false%

```

```

1162     \begingroup%

```

We get the next series of samewords data in the list of samewords data for the current edtext level. We push them inside `\sw@inthisedtext`.

```

1163     \ifledRcol%

```

```

1164     \ifcsundef{sw@list@edtextR@the\@edtext@level}%
1165         {\global\let\sw@inthisedtext\empty}%
1166         {\ifcsempy{sw@list@edtextR@the\@edtext@level}%
1167 {\global\let\sw@inthisedtext\empty}%
1168 {\expandafter\gl@p\csname sw@list@edtextR@the\@edtext@level\endcsname\to\sw@inthisedtext}%
1169 }%
1170     \else%
1171         \ifcsundef{sw@list@edtext@the\@edtext@level}%
1172             {\global\let\sw@inthisedtext\empty}%
1173             {\ifcsempy{sw@list@edtext@the\@edtext@level}%
1174 {\global\let\sw@inthisedtext\empty}%
1175 {\expandafter\gl@p\csname sw@list@edtext@the\@edtext@level\endcsname\to\sw@inthisedtext}%
1176 }%
1177     \fi%

```

`\@tag` Our normal lemma is just argument #1; but that argument could have further invocations of `\edtext` within it. We get a copy of the lemma without any `\edtext` macros within it by temporarily redefining `\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 `\edtext` restored; within this group we have also turned off the expansion of those control sequences commonly found within text that can cause trouble for us.

```

1178     \global\renewcommand{\@tag}{%
1179     \no@expands #1%
1180     }%

```

`\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`.

```

1181     \set@line%

```

`\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 `\edtext`. If we are in a right column (reledpar), we use `\insert@countR` instead of `\insert@count`.

```

1182     \ifledRcol \global\insert@countR \z@%
1183     \else      \global\insert@count \z@ \fi%

```

Now process the note-generating macros in argument #2 (i.e., `\Afootnote`, `\lemma`, etc.). `\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 `\ignorespaces` as well, to skip any spaces between macros when several are used in series.

```

1184     \ignorespaces #2\relax%

```

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

```

1185     \ifundefined{xpg@main@language}{%if not polyglossia
1186         \flag@start}%
1187     {\ifRTL\flag@end\else\flag@start\fi%
1188     }%

```

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

```

1189     \if@lemmacommand%
1190     \ifledRcol%
1191     \write\linenum@outR{\string\@lemma}%
1192     \else%
1193     \write\linenum@out{\string\@lemma}%
1194     \fi%
1195     \fi%

```

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

It is important that we generate and output all the notes for this chunk of text *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 #2 above, or in `\aftergroup` commands within that expansion.

```

1196     \endgroup%
1197     \showlemma{#1}%

```

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.

```

1198     \ifx\end@lemmas\empty \else%
1199     \gl@p\end@lemmas\to\x@lemma%
1200     \x@lemma%
1201     \global\let\x@lemma=\relax%
1202     \fi%
1203     \ifundefined{xpg@main@language}{%if not polyglossia
1204     \flag@end}%
1205     {%if@RTL\flag@start\else\flag@end\fi% With polyglossia, you must track whether the language reads
1206     }%

```

We switch to false some flags.

- The one that checks having footnotes inside a `\edtext`.
- The one that says we are inside a `\edtext`. In fact, it is not a flag, but a counter which is increased to 1 in each level of `\edtext`.
- The one that says we are inside a `\@lemma`.

```

1207     \global\@noneed@Footnotefalse%
1208     \global\advance\@edtext@level by -1%
1209     \global\@lemmacommand@false%

```

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

```

1210     \else%
1211     \showlemma{#1} (\textbf{\textsc{Edtext outside numbered paragraph}})\led@err@edtextoutsidestart%
1212     \fi%
1213 }%

```

```

1214
1215 \newcommand*{\flag@end}{%
1216   \ifledRcol%
1217     \write\linenum@outR{]}%
1218   \else%
1219     \write\linenum@out{]}%
1220   \fi}%
1221

```

`\ifnumberline` The `\ifnumberline` option can be set to FALSE to disable line numbering.

```

1222 \newif\ifnumberline
1223 \numberlinetrue

```

`\set@line` The `\set@line` macro is called by `\edtext` to put the line-reference field and font specifier for the current block of text into `\l@d@nums`.

One instance of `\edtext` may generate several notes, or it may generate none — it is legitimate for argument #2 to `\edtext` 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 is vital to also remove one and only one `\line@list` entry here.

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

```

1224 \newcommand*{\set@line}{%
1225   \ifledRcol
1226     \ifx\line@listR\empty
1227       \global\noteschanged@true
1228       \xdef\l@d@nums{000|000|000|000|000|000|\edfont@info}%
1229     \else
1230       \gl@p\line@listR\to\@tempb
1231       \xdef\l@d@nums{\@tempb|\edfont@info}%
1232       \global\let\@tempb=\undefined
1233     \fi
1234   \else
1235     \ifx\line@list\empty
1236       \global\noteschanged@true
1237       \xdef\l@d@nums{000|000|000|000|000|000|\edfont@info}%
1238     \else
1239       \gl@p\line@list\to\@tempb
1240       \xdef\l@d@nums{\@tempb|\edfont@info}%
1241       \global\let\@tempb=\undefined
1242     \fi
1243   \fi}
1244

```

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

```

1245 \newcommand*{\edfont@info}{\f@encoding/\f@family/\f@series/\f@shape}
1246

```

## VI.2 Substitute lemma

`\lemma` The `\lemma{<text>}` macro allows you to change the lemma that is passed on to the notes. Read about `\@tag` in normal `\edtext` macro for more details about `\sw@list@inedtext` and `\no@expands` (VI.1 p. 100).

```
1247 \unless\ifnocritical@
1248 \newcommand*{\lemma}[1]{%
1249   \global\@lemmacommand@true%
1250   \global\renewcommand{\@tag}{%
1251     \no@expands #1%
1252   }%
1253   \ignorespaces%
1254 }
```

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

```
1255 \newcommand{\@lemma}{%
1256   \booltrue{lemmacommand@the\@edtext@level}%
1257 }%
1258 \fi
```

`\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.

```
1259 \newif\if@lemmacommand@
```

## VI.3 Substitute line numbers

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

As argument `\linenum` takes a set of seven parameters separated by vertical bars, in the format used internally for `\l@d@nums` (see V.9 p. 77): 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 do not want to change, and you can omit a string of vertical bars at the end of the argument. Hence `\linenum{18|4|0|18|7|1|0}` is an invocation that changes all the parameters, but `\linenum{|3}` only changes the starting line number, and leaves the rest unaltered.

We use `\\` as an internal separator for the macro parameters.

```
1260 \newcommand*{\linenum}[1]{%
1261   \xdef\@tempa{#1|||||\\noexpand\\l@d@nums}%
1262   \global\let\l@d@nums=\empty
1263   \expandafter\line@set\@tempa|\\ignorespaces}
```

`\line@set` `\linenum` calls `\line@set` to do the actual work; it looks at the first number in the argument to `\linenum`, sets the corresponding value in `\l@d@nums`, and then calls itself to process the next number in the `\linenum` argument, if there are more numbers in `\l@d@nums` to process.

```

1264 \def\line@set#1|#2\\#3|#4\\{%
1265   \gdef\@tempb{#1}%
1266   \ifx\@tempb\empty
1267     \l@d@add{#3}%
1268   \else
1269     \l@d@add{#1}%
1270   \fi
1271   \gdef\@tempb{#4}%
1272   \ifx\@tempb\empty\else
1273     \l@d@add{|}\line@set#2\\#4\\%
1274   \fi}

```

`\l@d@add` `\line@set` uses `\l@d@add` to tack numbers or vertical bars onto the right hand end of `\l@d@nums`.

```

1275 \newcommand{\l@d@add}[1]{\xdef\l@d@nums{\l@d@nums#1}}
1276

```

## VI.4 Lemma disambiguation

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

The general mechanism is the following:

- **At the first run**, each `\sameword` command increments an etoolbox counter the name of which contains the argument of the `\sameword` commands.
- Then this counter, associated with the argument of `\sameword` is stored, with the `\@sw` command, in the auxiliary file of the current eldmac 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 subtract 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 \saweword 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 \saweword in first argument of \edtext.

For example, suppose a line with nested \edtexts which contains some word marked by \sameword and having the following relative rank:

bar<sup>1</sup> foo<sup>1</sup> foo<sup>2</sup> bar<sup>2</sup> foo<sup>3</sup> (A)(B) foo<sup>4</sup> bar<sup>3</sup> (C) foo<sup>5</sup> (D) bar<sup>4</sup> (E)

In this example, all lemma information for \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 \edtext.

The list for \edtexts of level 1 is  $\{\{1, 2, 2, 3, 4, 3\}, \{5, 4\}\}$ .

The list for \edtexts of level 2 is  $\{\{1, 2, 2, 3\}, \{5\}\}$ .

As you can see, the mandatory argument of \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 \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 \sameword command is let \sameword@inedtext.
  - At each call of this \sameword@inedtext, we step to the next element of the list associated to the note. Let it be  $r$ .

- For the word marked by `\sameword`, we calculate how many time it is called in its line. To do it:
  - \* We get the absolute line number of the current `\sameword`. This absolute line number was stored with list of relative rank for the current `\edtext`. That means, in the previous example, that, if the absolute line number of `\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 `\sameword` is required because a `\edtext` can be overlap many lines, but `\sameword` can't 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  $n$ .
- If  $n > 1$ , that mean the current word appears more than once time in its line. In this case, we call `\showwordrank` with the word as first argument and  $r$  as second argument. If the word is called only once, we just print it.

After theory, implementation.

`\get@sw@txt` 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`.<sup>25</sup>

Because there is a bug with `\detokenize` and  $X_{\text{TeX}}$  when using non BMP characters<sup>26</sup>, we detokenize only for not  $X_{\text{TeX}}$  engines. In any case, in  $X_{\text{TeX}}$ , 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.

```

1277 \newcommand{\get@sw@txt}[1]{%
1278   \ifxetex%
1279     \xdef\sw@txt{#1}%
1280   \else%
1281     \expandafter\xdef\expandafter\sw@txt\expandafter{\detokenize{#1}}%
1282   \fi%
1283 }%
```

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

```

1284 \newcommandx{\sameword}[2][1,usedefault]{%
1285   \leavevmode%
1286   \get@sw@txt{#2}%
```

Now, the real code. First, increment the counter corresponding to the argument.

```

1287 \unless\ifledRcol%
1288   \csnumgdef\sw@\sw@txt{\csuse{\sw@\sw@txt}+1}%

```

Then, write its value to the numbered file.

```

1289   \protected@write\linenumout{\string\sw@\sw@txt}{\csuse{\sw@\sw@txt}{#1}}%
```

<sup>25</sup>See <http://tex.stackexchange.com/q/244538/7712>.

<sup>26</sup><http://sourceforge.net/p/xetex/bugs/108/>

Do the same thing if we are in the right columns.

```

1290 \else%
1291   \csnumgdef{sw@\sw@txt@R}{\csuse{sw@\sw@txt@R}+1}%
1292   \protected@write\linenum@outR{\string\sw@\sw@txt}{\csuse{sw@\sw@txt@R}}{#1}}%
1293 \fi%
```

And print the word.

```

1294 #2%
1295 }%
```

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

\if@addsw

```

1296 \newif\if@addsw%
```

\@sw The command printed in the auxiliary files.

```

1297 \newcommand{\@sw}[3]{%
1298   \get@sw@txt{#1}%
1299   \unless\ifledRcol%
```

First, define a counter which store the second argument as value for a each paired absolute line number/first argument

```

1300   \csxdef{sw@\sw@txt @\the\absline@num @\the\section@num}{#2}%
```

If such argument was not defined for the preceding line, define it.

```

1301   \numdef{\prev@line}{\the\absline@num-1}%
1302   \ifcsundef{sw@\sw@txt @\prev@line @\the\section@num}{%
1303     \csnumgdef{sw@\sw@txt @\prev@line @\the\section@num}{#2-1}%
1304   }{}}%
```

Then, calculate the position of the word in the line.

```

1305   \numdef{\the@sw}{#2-\csuse{sw@\sw@txt @\prev@line @\the\section@num}}%
```

And do the same thing for the right side.

```

1306 \else%
1307   \csxdef{sw@\sw@txt @\the\absline@numR @\the\section@numR @R}{#2}%
1308   \numdef{\prev@line}{\the\absline@numR-1}%
1309   \ifcsundef{sw@\sw@txt @\prev@line @\the\section@numR @R}{%
1310     \csnumgdef{sw@\sw@txt @\prev@line @\the\section@numR @R}{#2-1}%
1311   }{}}%
1312   \numdef{\the@sw}{#2-\csuse{sw@\sw@txt @\prev@line @\the\section@numR @R}}%
1313 \fi%
```

And now, add it to the list of \@sw for the current edtext, in all depth.

```

1314 \@tempcnta=\@edtext@level
1315 \@whilenum{\@tempcnta>0}\do{%
1316   \ifcsdef{sw@list@edtext@tmp@\the\@tempcnta}%
1317   {%
1318     \@addswfalse%
1319     \notbool{lemmacommand@\the\@tempcnta}%
1320     {\@addswtrue}%
```

```

1321      {\IfStrEq{#3}{inlemma}%
1322       {\@addswtrue}%
1323       {%
1324        \def\do##1{%
1325         \ifnumequal{##1}{\the\@tempcnta}%
1326         {\@addswtrue\listbreak}%
1327         }%
1328       }%
1329       \docsvlist{#3}%
1330     }%
1331   }%
1332   \if@addsw%
1333     \letcs{\@tmp}{sw@list@edtext@tmp@\the\@tempcnta}%
1334     \ifledRcol%
1335       \xright@appenditem{\the@sw}{\the\absline@numR}}\to\@tmp%
1336     \else%
1337       \xright@appenditem{\the@sw}{\the\absline@num}}\to\@tmp%
1338     \fi%
1339     \cslet{sw@list@edtext@tmp@\the\@tempcnta}{\@tmp}%
1340   \fi%
1341 }%
1342 {}%
1343 \advance\@tempcnta by -1%
1344 }%
1345 }%

```

\sameword@inedtext The command called when \sameword is called in a \edtext.

```

1346 \newcommandx{\sameword@inedtext}[2][1,usedefault]{%
1347   \get@sw@txt{#2}%
1348   \unless\ifledRcol%

```

Just a precaution.

```

1349   \ifx\sw@list@inedtext\empty%
1350     \def\the@sw{999}%
1351     \def\this@absline{-99}%
1352   \else%

```

But in many cases, at this step, we should have some content in the list \sw@list@inedtext, which contains the reference for \edtext.

```

1353     \gl@p\sw@list@inedtext\to\@tmp%
1354     \edef\the@sw{\expandafter\@firstoftwo\@tmp}%
1355     \edef\this@absline{\expandafter\@secondoftwo\@tmp}%
1356   \fi%

```

First, calculate the number of occurrences of the word in the current line

```

1357     \ifcsdef{sw@\sw@txt @\this@absline @\the\section@num}{%
1358       \numdef{\prev@line}{\this@absline-1}%
1359       \numdef{\sw@atthisline}{\csuse{sw@\sw@txt @\this@absline @\the\section@num}-\csuse{sw@
1360       }%
1361       {\numdef{\sw@atthisline}{0}}%

```

Finally, print the rank, but only if there is more than one occurrence of the word in the current line.

```
1362     \ifnumgreater{\sw@atthisline}{1}%
1363         {\showwordrank{#2}{\the@sw}}%
1364         {#2}%
```

And the same for right side.

```
1365     \else%
1366         \ifx\sw@list@inedtext\empty%
1367             \def\the@sw{999}%
1368             \def\this@absline{-99}%
1369         \else%
1370             \gl@p\sw@list@inedtext\to\@tmp%
1371             \edef\the@sw{\expandafter\@firstoftwo\@tmp}%
1372             \edef\this@absline{\expandafter\@secondoftwo\@tmp}%
1373         \fi%
1374         \ifcsdef{sw@\sw@txt @\this@absline @\the\section@numR @R}{%
1375             \numdef{\prev@line}{\this@absline-1}%
1376             \numdef{\sw@atthisline}{\csuse{sw@\sw@txt @\this@absline @\the\section@numR @R}-\csuse{sw@\sw@txt
1377             }%
1378             {\numdef{\sw@atthisline}{0}}%
1379             \ifnumgreater{\sw@atthisline}{1}%
1380                 {\showwordrank{#2}{\the@sw}}%
1381                 {#2}%
1382         \fi%
1383 }%
```

\showwordrank

```
1384 % Finally, the way the rank will be printed.
1385 \newcommand{\showwordrank}[2]{%
1386     #1\textsuperscript{#2}%
1387 }%
```

## VII 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.

### VII.1 Boxes, counters, \pstart and \pend

<pre>\raw@text \ifnumberedpar@ \numberedpar@true \numberedpar@false \num@lines \one@line \par@line</pre>	<p>Here are numbers and flags that are used internally in the course of the paragraph decomposition.</p>
--	--

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 is 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.

```
1388 \newbox\raw@text
1389 \newif\ifnumberedpar@
1390 \newcount\num@lines
1391 \newbox\one@line
1392 \newcount\par@line
```

<pre>\pstart \AtEveryPstart \numberpstarttrue \numberpstartfalse \labelpstarttrue \labelpstartfalse \thepstart</pre>	<p><code>\pstart</code> starts the paragraph by clearing the <code>\inserts@list</code> list and other relevant variables, and then arranges for the subsequent text to go into the <code>\raw@text</code> box.</p> <p><code>\pstart</code> needs to appear at the start of every paragraph that is to be numbered; the <code>\autopar</code> command below may be used to insert these commands automatically.</p> <p>Beware: everything that occurs between <code>\pstart</code> and <code>\pend</code> is happening within a group; definitions must be global if you want them to survive past the end of the paragraph.</p>
--	--

```
1393
1394 \newcommand{\AtEveryPstart}[1]{%
1395   \ifstrempy{#1}%
1396   {\xdef\at@every@pstart{}}%
1397   {\gdef\at@every@pstart{\noindent#1}}%
1398 }%
1399 \xdef\at@every@pstart{}%
1400
1401 \newcounter{pstart}
1402 \renewcommand{\thepstart}{\bfseries\@arabic@c@pstart}. }
1403 \newif\ifnumberpstart
1404 \numberpstartfalse
1405 \newif\iflabelpstart
1406 \labelpstartfalse
1407 \newcommand*{\pstart}[1][1]{%
1408   \normal@pars%
1409   \ifstrempy{#1}{\at@every@pstart}{\noindent#1}%
1410   \ifautopar%
1411     \autopar%
1412   \fi%
1413   \ifluatex%
1414     \edef\l@luatexttextdir@L{\the\luatexttextdir}%
1415   \fi%
1416   \if@nobreak%
1417     \let\@oldnobreak\@nobreaktrue%
1418   \else%
1419     \let\@oldnobreak\@nobreakfalse%
1420   \fi%
1421   \@nobreaktrue%
```

```

1422 \ifnumbering \else%
1423   \led@err@PstartNotNumbered%
1424   \beginnumbering%
1425 \fi%
1426 \ifnumberedpar@%
1427   \led@err@PstartInPstart%
1428   \pend%
1429 \fi%
1430 \list@clear{\inserts@list}%
1431 \global\let\next@insert=\empty%
1432 \begingroup\normal@pars%
1433 \global\advance \l@dnumpstartsL\@one
1434 \global\setbox\raw@text=\vbox\bgroup%
1435   \ifautopar\else%
1436     \ifnumberpstart%
1437       \ifinstanza\else%
1438         \ifsidepstartnum\else%
1439           \thepstart%
1440         \fi%
1441       \fi%
1442     \fi%
1443   \fi%
1444   \numberedpar@true%
1445   \iflabelpstart\protected@edef\@currentlabel%
1446     {\p@pstart\thepstart}%
1447   \fi%
1448   \l@dzeropenalties%
1449 }

```

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

```

1450 \newcommand*{\pend}[1][1]{\ifnumbering \else%
1451   \led@err@PendNotNumbered%
1452 \fi%
1453 \global\l@dskipversenumberfalse%
1454 \ifnumberedpar@ \else%
1455   \led@err@PendNoPstart%
1456 \fi%

```

We set all the usual interline penalties to zero and then immediately call \endgraf to end the paragraph; this ensures that there will 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 \vbox ends. Then we call \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 are not any more lines left.

```

1457 \l@dzeropenalties%
1458 \endgraf\global\num@lines=\prevgraf\egroup%
1459 \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. We can't reset line number at the beginning of \pstart, as

`\setline` is parsed at the end of previous `\pend`, and so, we must do it at the end of first line of `pstart`.

```

1460 \csnumdef{pstartline}{0}%
1461 \loop\ifvbox\raw@text%
1462   \csnumdef{pstartline}{\pstartline+1}%
1463   \do@line%
1464   \ifbypstart%
1465     \ifnumequal{\pstartline}{1}{\setline{1}\resetprevline@}{}%
1466     \fi%
1467 \repeat%
```

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

```

1468 \flush@notes%
1469 \endgroup%
1470 \ignorespaces%
```

Increase `pstart` counter.

```

1471 \ifnumberpstart%
1472   \pstartnumtrue%
1473 \fi%
1474 \addtocounter{pstart}{1}%
```

Restore paragraph, nobreak setting and autopar setting.

```

1475 \normal@pars%
1476 \@oldnobreak%
1477 \ifautopar%
1478   \autopar%
1479 \fi%
```

Print the optional argument of `\pend` or the content printed after every `\pend`

```

1480 \ifstrepty{#1}{\at@every@pend}{\noindent#1}%
1481 }
1482
```

Here, two macros to insert content after every `\pend`, between numbered line. `\AtEveryPend` is the user macro, `\at@every@pend` is macro set by it.

```

\AtEveryPend
\at@every@pend 1483
1484 \newcommand{\AtEveryPend}[1]{%
1485   \ifstrepty{#1}%
1486     {\xdef\at@every@pend{}}%
1487     {\gdef\at@every@pend{\noindent#1}}%
1488 }%
1489 \xdef\at@every@pend{}%
1490
```

`\l@dzero penalties` A macro to zero penalties for `\pend` or `\pstart`.

```

1491 \newcommand*\l@dzero penalties{%
1492   \brokenpenalty \z@ \clubpenalty \z@
1493   \displaywidowpenalty \z@ \interlinepenalty \z@ \predisplaypenalty \z@
```



```

1494 \postdisplaypenalty \z@ \widowpenalty \z@}
1495

```

`\autopar` In most cases it is 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 `\vbox` that `\pstart` creates, and the rest of the paragraph will not be numbered. Such paragraphs need to be started explicitly using `\indent`, `\noindent`, or `\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 `\everypar`: we do not 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 `\lastbox` and save the width, and then skip backwards over the `\parskip` that has been added for this paragraph. Then we start again with `\pstart`, restoring the indentation that we saved, and locally change `\par` so that it will do our `\pend` for us.

```

1496 \newif\ifautopar
1497 \autoparfalse
1498 \newcommand*{\autopar}{
1499   \ifledRcol
1500     \ifnumberingR \else
1501       \led@err@AutoparNotNumbered
1502       \beginnumberingR
1503     \fi
1504   \else
1505     \ifnumbering \else
1506       \led@err@AutoparNotNumbered
1507       \beginnumbering
1508     \fi
1509   \fi
1510   \autopartrue
1511   \everypar{\setbox0=\lastbox
1512     \endgraf \vskip-\parskip
1513     \pstart \noindent \kern\wd0 \ifnumberpstart\ifinstanza\else\thepstart\fi\fi
1514     \let\par=\pend}%
1515   \ignorespaces}

```

`\normal@pars` 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 will want to do this within a footnotes, for example.

```
1516 \newcommand*{\normal@pars}{\everypar{}\let\par\endgraf}
1517
```

`\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.

```
1518 \newif\ifautopar@pause
```

## VII.2 Processing one line

### VII.2.1 General process

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

```
1519 \newcommand*{\l@dunhbox@line}[1]{\unhbox #1}
1520 \newcommand*{\do@line}{%
1521   {\vbadness=10000
1522     \splittopskip=\z@
1523     \do@linehook
1524     \l@emptyd@ta
1525     \global\setbox\one@line=\vsplit\raw@text to\baselineskip}%
1526     \unvbox\one@line \global\setbox\one@line=\lastbox
1527     \getline@num
1528     \IfStrEq{\led@pb@setting}{before}{\led@check@pb\led@check@nopb}{%
1529       \ifnum\@lock>\@ne
1530         \inserthangingsymboltrue
1531       \else
1532         \inserthangingsymbolfalse
1533       \fi
1534       \check@pb@in@verse
1535       \ifl@dhidenumber%
1536         \global\l@dhidenumberfalse%
1537         \f@x@locks%
1538       \else%
1539         \affixline@num%
1540       \fi%
1541       \xifinlist{\the\l@dnumpstargsL}{\eled@sections@}%
1542         {\print@eledsection}%
1543         {\print@line}%
1544       \IfStrEq{\led@pb@setting}{after}{\led@check@pb\led@check@nopb}{%
1545         }%
1546     }
```

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

**VII.2.2 Process for “normal” line**

```

\print@line \print@line is for normal line, i. e line without sectioning command.
1546 \def\print@line{
  Insert the pstart number in side, if we are in the first line of a pstart.
1547   \affixpstart@num%
  The line will be boxed, to have the good width.
1548   \hb@xt@ \linewidth{%
    User hook.
1549   \do@insidelinehook%
    Left line number
1550   \l@dld@ta%
    Restore marginal and footnotes.
1551   \add@inserts\affixside@note%
    Print left notes.
1552   \l@dlsn@te
    Boxes the line, writes information about new line in the numbered file.
1553   {\ledllfill\hb@xt@ \wd\one@line{\new@line%
    If we use LuaℒTeX then restore the direction.
1554   \ifluatex%
1555     \luatextextdir\l@luatextextdir@L%
1556     \fi%
    Insert, if needed, the hanging symbol.
1557     \inserthangingsymbol %Space kept for backward compatibility
    And so, print the line.
1558     \l@dunhbox@line{\one@line}}%
    Right line number
1559     \ledrlfill\l@drd@ta%
    Print right notes.
1560     \l@drsn@te
1561     }}%
    And reinsert penalties (for page breaking)...
1562     \add@penalties%
1563 }

```

### VII.2.3 Process for line containing \eledsection command

`\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.

```

1564 \def\print@eledsection{%
1565     \add@inserts\affixside@note%
1566     \numdef{\temp@}{\l@dnumpstartsL-1}%
1567     \xifinlist{\temp@}{\eled@sections@@}{\@nbreaktrue}{\@nbreakfalse}%
1568     \@eled@sectioningtrue%
1569     \csuse{eled@sectioning@the\l@dnumpstartsL}%
1570     \@eled@sectioningfalse%
1571     \global\csundef{eled@sectioning@the\l@dnumpstartsL}%
1572     \if@RTL%
1573         \hspace{-3\paperwidth}%
1574     {\hbox{\l@dunhbox@line{\one@line}} \new@line}%
1575     \else%
1576         \hspace{3\paperwidth}%
1577     {\new@line \hbox{\l@dunhbox@line{\one@line}}}%
1578     \fi%
1579     \vskip-\baselineskip%
1580 }
```

### VII.2.4 Hooks

`\do@linehook` `\do@linehook` 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 not.

```

1581 \newcommand*{\do@linehook}{}
1582 \newcommand*{\do@insidelinehook}{}

```

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

```

1583 \newcommand*{\dolinehook}[1]{\gdef\do@linehook{#1}}%
1584 \newcommand*{\doinsidelinehook}[1]{\gdef\do@insidelinehook{#1}}%
1585

```

### VII.2.5 Sidenotes and marginal line number initialization

`\l@emptyd@ta` `\l@emptyd@ta` Nulls the `\l@emptyd@ta`, which may later hold line numbers. Similarly for `\l@dcsnotetext`, `\l@dld@ta` `\l@dcsnotetext@l`, `\l@dcsnotetext@r` for the texts of the sidenotes, left and right notes.

```

1586 \newcommand*{\l@emptyd@ta}{}%
1587 \gdef\l@dld@ta{}%
1588 \gdef\l@drd@ta{}%
1589 \gdef\l@dcsnotetext@l{}%
1590 \gdef\l@dcsnotetext@r{}%

```

```

1591 \gdef\l@dcstotetext{}}
1592

```

`\l@dlsn@te` Zero width boxes of the left and right side notes, together with their kerns.

```

\l@drsn@te 1593 \newcommand{\l@dlsn@te}{%
1594 \hb@xt@ \z@{\hss\box\l@dlp@rbox\kern\ledlsnotesep}}
1595 \newcommand{\l@drsn@te}{%
1596 \hb@xt@ \z@{\kern\ledrsnotesep\box\l@drp@rbox\hss}}
1597

```

`\ledllfill` These macros are called at the left (`\ledllfill`) and the right (`\ledllfill`) of each  
`\ledrlfill` numbered line. The initial definitions correspond to the original code for `\do@line`.

```

1598 \newcommand*{\ledllfill}{\hfil}
1599 \newcommand*{\ledrlfill}{\hfil}
1600

```

## VIII Line and page number computation

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

```

1601 \newcommand*{\getline@num}{%
1602 \global\advance\absline@num \@ne%
1603 \do@actions
1604 \do@ballast
1605 \ifnumberline
1606 \ifsublines@
1607 \ifnum\sub@lock<\tw@
1608 \global\advance\subline@num \@ne
1609 \fi
1610 \else
1611 \ifnum\@lock<\tw@
1612 \global\advance\line@num \@ne
1613 \global\subline@num \z@
1614 \fi
1615 \fi
1616 \fi
1617 }

```

`\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,  $\text{\TeX}$  will be given extra encouragement to break the page here (see XI.2 p. 126).

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

```

1618 \newcount\ballast@count
1619 \newcounter{ballast}
1620 \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.

```

1621 \newcommand*{\do@ballast}{\global\ballast@count \z@
1622 \begingroup
1623   \advance\absline@num \@ne
1624   \ifnum\next@actionline=\absline@num
1625     \ifnum\next@action>-1001\relax
1626       \global\advance\ballast@count by -\c@ballast
1627     \fi
1628   \fi
1629 \endgroup}

```

`\do@actions` The `\do@actions` macro looks at the list of actions to take at particular absolute line numbers, and does everything that is 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 is just `\relax`.

```

1630 \newcommand*{\do@actions}{%
1631   \global\let\do@actions@next=\relax
1632   \ifnum\absline@num<\next@actionline\else

```

First, page number changes, which will generally be the most common actions. If we are restarting lineation on each page, this is where it happens.

```

1633     \ifnum\next@action>-1001
1634       \global\page@num=\next@action
1635       \ifbypage@
1636         \global\line@num=\z@ \global\subline@num=\z@
1637         \resetprevline@
1638       \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`.)

```

1639     \else
1640       \ifnum\next@action<-4999
1641         \@l@dttempcnta=-\next@action
1642         \advance\@l@dttempcnta by -5001
1643         \ifsublines@
1644           \global\subline@num=\@l@dttempcnta
1645         \else
1646           \global\line@num=\@l@dttempcnta
1647         \fi

```

We rescale the value in `\@l@dttempcnta` so that we can use a case statement.

```

1648     \else

```

```

1649         \@l@dttempcnta=-\next@action
1650         \advance\@l@dttempcnta by -1000
1651         \do@actions@fixedcode
1652     \fi
1653 \fi

```

Now we get information about the next action off the list, and then set `\do@actions@next` so that we will call ourself recursively: the next action might also be for this line.

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

```

1654     \ifx\actionlines@list\empty
1655         \gdef\next@actionline{1000000}%
1656     \else
1657         \gl@p\actionlines@list\to\next@actionline
1658         \gl@p\actions@list\to\next@action
1659         \global\let\do@actions@next=\do@actions
1660     \fi
1661 \fi

```

Make the recursive call, if necessary.

```

1662 \do@actions@next}
1663

```

`\do@actions@fixedcode` This macro handles the fixed codes for `\do@actions`. It is one big case statement.

```

1664 \newcommand*{\do@actions@fixedcode}{%
1665     \ifcase\@l@dttempcnta
1666         \or% % 1001
1667             \global\sublines@true
1668         \or% % 1002
1669             \global\sublines@false
1670         \or% % 1003
1671             \global\@lock=\@ne
1672         \or% % 1004
1673             \ifnum\@lock=\tw@
1674                 \global\@lock=\thr@@
1675             \else
1676                 \global\@lock=\z@
1677             \fi
1678         \or% % 1005
1679             \global\sub@lock=\@ne
1680         \or% % 1006
1681             \ifnum\sub@lock=\tw@
1682                 \global\sub@lock=\thr@@
1683             \else
1684                 \global\sub@lock=\z@
1685             \fi
1686         \or% % 1007
1687             \l@dskipnumbertrue
1688         \or% % 1008
1689             \l@dskipversenumbertrue%

```

```

1690 \or%      % 1009
1691      \l@dhiddenumbertrue
1692 \else
1693      \led@warn@BadAction
1694 \fi}
1695
1696

```

## IX Line number printing

`\affixline@num` `\affixline@num` 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:

$$n = \text{int}((\text{linenum} - \text{firstlinenum}) / \text{linenumincrement})$$

$$m = \text{firstlinenum} + (n \times \text{linenumincrement})$$

(where *int* truncates a real number to an integer). *m* will be equal to *linenum* only if we are 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 `\line@num ≤ \firstlinenum`, we compare the two directly instead of making these calculations.

We compute, in the scratch counter `\@l@tempcnta`, 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@tempcntb` for comparison.

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

```

1697 \newcommand*{\affixline@num}{%

```

No number is attached if `\ifl@dskipnumber` 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).

```

1698 \ifledgroupnotesL@else
1699 \ifnumberline
1700 \ifl@dskipnumber
1701 \global\l@dskipnumberfalse
1702 \else
1703 \ifsublines@
1704 \l@tempcntb=\subline@num
1705 \ifnum\subline@num>\c@firstsublinenum
1706 \l@tempcnta=\subline@num
1707 \advance\l@tempcnta by-\c@firstsublinenum
1708 \divide\l@tempcnta by\c@sublinenumincrement
1709 \multiply\l@tempcnta by\c@sublinenumincrement
1710 \advance\l@tempcnta by\c@firstsublinenum
1711 \else
1712 \l@tempcnta=\c@firstsublinenum
1713 \fi

```



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.

```
1714      \ch@cksub@l@ck
```

Now the line number case, which works the same way.

```
1715      \else
```

```
1716      \@l@tempcntb=\line@num
```

Check on the `\linenumberlist` If it is `\empty` use the standard algorithm.

```
1717      \ifx\linenumberlist\empty
```

```
1718      \ifnum\line@num>\c@firstlinenum
```

```
1719      \@l@tempcnta=\line@num
```

```
1720      \advance\@l@tempcnta by-\c@firstlinenum
```

```
1721      \divide\@l@tempcnta by\c@linenumincrement
```

```
1722      \multiply\@l@tempcnta by\c@linenumincrement
```

```
1723      \advance\@l@tempcnta by\c@firstlinenum
```

```
1724      \else
```

```
1725      \@l@tempcnta=\c@firstlinenum
```

```
1726      \fi
```

```
1727      \else
```

The `\linenumberlist` was not `\empty`, so here is Wayne's numbering mechanism.

This takes place in  $\TeX$ 's mouth.

```
1728      \@l@tempcnta=\line@num
```

```
1729      \edef\rem@inder{\,\linenumberlist,\number\line@num,}%
```

```
1730      \edef\sc@n@list{\def\noexpand\sc@n@list
```

```
1731      ####1,\number\@l@tempcnta,####2|\def\noexpand\rem@inder{####2}}}%
```

```
1732      \sc@n@list\expandafter\sc@n@list\rem@inder|%
```

```
1733      \ifx\rem@inder\empty%
```

```
1734      \advance\@l@tempcnta\@ne
```

```
1735      \fi
```

```
1736      \fi
```

A locking check for lines, just like the version for sub-line numbers above.

```
1737      \ch@ck@l@ck
```

```
1738      \fi
```

The following tests are true if we need to print a line number.

```
1739      \ifnum\@l@tempcnta=\@l@tempcntb
```

```
1740      \ifl@dskipversenumber\else
```

If we got here, we are going to print a line number; so now we need to calculate a number that will tell us which side of the page will get the line number. We start from `\line@margin`, which asks for one side always if it is less than 2; and then if the side does depend on the page number, we simply add the page number to this side code—because the values of `\line@margin` have been devised so that this produces a number that is even for left-margin numbers and odd for right-margin numbers.

For  $\mathbb{E}\mathbb{T}\mathbb{X}$  we have to consider two column documents as well. In this case Peter Wilson thought 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 `twocolumn` stuff before going on with the original code.

\l@dld@ta A left line number is stored in \l@dld@ta and a right one in \l@drd@ta.

```

\l@drd@ta 1741          \if@twocolumn
1742          \if@firstcolumn
1743          \gdef\l@dld@ta{\llap{\leftlinenum}}}%
1744          \else
1745          \gdef\l@drd@ta{\rlap{\rightlinenum}}}%
1746          \fi
1747        \else
1748        \@l@tempcntb=\line@margin
1749        \ifnum\@l@tempcntb>\@ne
1750        \advance\@l@tempcntb \page@num
1751        \fi
1752        \ifodd\@l@tempcntb
1753        \gdef\l@drd@ta{\rlap{\rightlinenum}}}%
1754        \else
1755        \gdef\l@dld@ta{\llap{\leftlinenum}}}%
1756        \fi
1757      \fi
1758    \fi
1759  \fi

```

Now fix the lock counters, if necessary. A value of 1 is advanced to 2; 3 advances to 0; other values are unchanged.

```

1760    \f@x@l@cks
1761    \fi
1762  \fi
1763 \fi
1764 }
1765

```

\ch@cksub@l@ck These macros handle line number locking for \affixline@num. \ch@cksub@l@ck  
\ch@ck@l@ck checks subline locking. If it fails, then we disable the line-number display by setting the  
\f@x@l@cks counters to arbitrary but unequal values.

```

1766 \newcommand*{\ch@cksub@l@ck}{%
1767   \ifcase\sub@lock
1768   \or
1769   \ifnum\sublock@disp=\@ne
1770   \@l@tempcntb=\z@ \@l@tempcnta=\@ne
1771   \fi
1772   \or
1773   \ifnum\sublock@disp=\tw@ \else
1774   \@l@tempcntb=\z@ \@l@tempcnta=\@ne
1775   \fi
1776   \or
1777   \ifnum\sublock@disp=\z@
1778   \@l@tempcntb=\z@ \@l@tempcnta=\@ne
1779   \fi
1780 \fi}

```

Similarly for line numbers.

```

1781 \newcommand*{\ch@ck@l@ck}{%
1782   \ifcase\@lock
1783     \or
1784       \ifnum\lock@disp=\@ne
1785         \@l@dttempcntb=\z@ \@l@dttempcnta=\@ne
1786       \fi
1787     \or
1788       \ifnum\lock@disp=\tw@ \else
1789         \@l@dttempcntb=\z@ \@l@dttempcnta=\@ne
1790       \fi
1791     \or
1792       \ifnum\lock@disp=\z@
1793         \@l@dttempcntb=\z@ \@l@dttempcnta=\@ne
1794       \fi
1795   \fi}

```

Fix the lock counters. A value of 1 is advanced to 2; 3 advances to 0; other values are unchanged.

```

1796 \newcommand*{\f@x@l@cks}{%
1797   \ifcase\@lock
1798     \or
1799       \global\@lock=\tw@
1800     \or \or
1801       \global\@lock=\z@
1802     \fi
1803   \ifcase\sub@lock
1804     \or
1805       \global\sub@lock=\tw@
1806     \or \or
1807       \global\sub@lock=\z@
1808     \fi}
1809

```

## X 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 is tried in the \leftpstartnum and \rightstartnum commands. After the try, it is set to FALSE.
- \leftpstartnum  
\rightstartnum  
\ifsidepstartnum

```

1810
1811 \newif\ifsidepstartnum
1812 \newcommand*{\affixpstart@num}{%
1813   \ifsidepstartnum
1814     \if@twocolumn
1815       \if@firstcolumn
1816         \gdef\l@dld@ta{\llap{\leftpstartnum}}}%
1817       \else
1818         \gdef\l@drd@ta{\rlap{\rightpstartnum}}}%
1819     \fi
1820   \else
1821     \@l@dttempcntb=\line@margin
1822     \ifnum\@l@dttempcntb>\@ne
1823       \advance\@l@dttempcntb \page@num
1824     \fi
1825     \ifodd\@l@dttempcntb
1826       \gdef\l@drd@ta{\rlap{\rightpstartnum}}}%
1827     \else
1828       \gdef\l@dld@ta{\llap{\leftpstartnum}}}%
1829     \fi
1830   \fi
1831 \fi
1832
1833 }
1834 %
1835
1836 \newif\ifpstartnum
1837 \pstartnumtrue
1838 \newcommand*{\leftpstartnum}{%
1839   \ifpstartnum\thepstart
1840   \kern\linenumsep\fi
1841   \global\pstartnumfalse
1842 }
1843 \newcommand*{\rightpstartnum}{%
1844   \ifpstartnum
1845   \kern\linenumsep
1846   \thepstart
1847   \fi
1848   \global\pstartnumfalse
1849 }

```

## XI Restoring footnotes and penalties

Because of the paragraph decomposition process in order to number line, reledmac must hack the standard way T<sub>E</sub>X works in order to manage insertion of footnotes, both critical and familiar.

We need to call the `\insert` commands not when the content of `\pstart...pend` is read by T<sub>E</sub>X but when each individual line is typeset.

Consequently, when reading the content of `\pstart...\pend`, we store the insertion (footnotes) in an specific `reledmac`'s list, and we restore them to the vertical list when printing each individual line.

## XI.1 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.

```
1850 \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.

`\add@inserts@next`

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 is just `\relax`.

```
1851 \newcommand*{\add@inserts}{%
```

```
1852 \global\let\add@inserts@next=\relax
```

If `\inserts@list` is empty, there are not any more notes or insertions for this paragraph, and we need not waste our time.

```
1853 \ifx\inserts@list\empty \else
```

The `\next@insert` macro records the number of the line that receives the next footnote or other insert; it is empty when we start out, and just after we have affixed a note or insert.

```
1854 \ifx\next@insert\empty
```

```
1855 \ifx\insertlines@list\empty
```

```
1856 \global\noteschanged@true
```

```
1857 \gdef\next@insert{100000}%
```

```
1858 \else
```

```
1859 \gl@p\insertlines@list\to\next@insert
```

```
1860 \fi
```

```
1861 \fi
```

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 will call ourself recursively: there might be another insert for this same line.

```
1862 \ifnum\next@insert=\absline@num
```

```
1863 \gl@p\inserts@list\to\@insert
```

```
1864 \@insert
```

```
1865 \global\let\@insert=\undefined
```

```
1866 \global\let\next@insert=\empty
```

```
1867 \global\let\add@inserts@next=\add@inserts
```

```
1868 \fi
```

```
1869 \fi
```

Make the recursive call, if necessary.

```
1870 \add@inserts@next}
```

```
1871
```

## XI.2 Penalties

`\add@penalties` `\add@penalties` 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 are working on at the moment. The count `\@l@tempcnta` 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 (VIII p. 117). Finally, the penalty is checked to see that it does not go below  $-10000$ .

```

1872 \newcommand*{\add@penalties}{\@l@tempcnta=\ballast@count
1873   \ifnum\num@lines>\@ne
1874     \global\advance\par@line \@ne
1875     \ifnum\par@line=\@ne
1876       \advance\@l@tempcnta \clubpenalty
1877     \fi
1878     \@l@tempcntb=\par@line \advance\@l@tempcntb \@ne
1879     \ifnum\@l@tempcntb=\num@lines
1880       \advance\@l@tempcnta \widowpenalty
1881     \fi
1882     \ifnum\par@line<\num@lines
1883       \advance\@l@tempcnta \interlinepenalty
1884     \fi
1885   \fi
1886   \ifnum\@l@tempcnta=\z@
1887     \relax
1888   \else
1889     \ifnum\@l@tempcnta>-10000
1890       \penalty\@l@tempcnta
1891     \else
1892       \penalty -10000
1893     \fi
1894   \fi}
1895
```

## XI.3 Printing leftover notes

`\flush@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 previous run of  $\TeX$ , then there can be leftover notes that have not yet been printed. An appropriate error message will be printed elsewhere; but it is best to go ahead and print these notes somewhere, even if it is 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 is not too far from the proper location, to which they will move on the next run.

```

1896 \newcommand*{\flush@notes}{%

```

```

1897 \xloop
1898   \ifx\inserts@list\empty \else
1899     \gl@p\inserts@list\to\@insert
1900     \@insert
1901     \global\let\@insert=\undefined
1902 \repeat}
1903

```

`\xloop` `\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 ... \if ... \else ... \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 *TUGboat* **8** (1987), pp. 184–5.

```

1904 \def\xloop#1\repeat{%
1905   \def\body{#1\expandafter\body\fi}%
1906   \body}
1907

```

## XII 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 many separate levels of the footnotes, not just one; and there are options to reformat footnotes into paragraphs or into multiple columns.

### XII.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` `\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.

```

1908 \def\select@lemmafont#1|#2|#3|#4|#5|#6|#7|{\select@lemmafont#7|}
1909 \def\select@lemmafont#1/#2/#3/#4|{%
1910   {\fontencoding{#1}\fontfamily{#2}\fontseries{#3}\fontshape{#4}%
1911   \selectfont}
1912

```

## XII.2 Individual note options

`\footnoteoptions@` The `\footnoteoption@` [*side*] [*options*] [*value*] changes the value of on options of Xfootnote, to switch between true and false.

```

1913 \newcommandx*\footnoteoptions@[3] [1=L,usedefault]{%
1914   \def\do##1{%
1915     \ifstrequal{#1}{L}{% In Leftside
1916       \xright@appenditem{\global\noexpand\settoggle{##1@}{#3}}\to\inserts@list% Switch too
1917       \global\advance\insert@count \@ne% Increment the left insert counter.
1918     }%
1919   {%
1920     \xright@appenditem{\global\noexpand\settoggle{##1@}{#3}}\to\inserts@listR% Switch t
1921     \global\advance\insert@countR \@ne% Increment the right insert counter insert.
1922   }%
1923 }%
1924 \notblank{#2}{\docsvlist{#2}}}% Parsing all options
1925 }
```

## XII.3 Notes language

`\footnotelang@lua` `\footnotelang@lua` is called to remember the information about the direction of a lemma when Lua<sub>TeX</sub> is used.

```

1926 \newcommandx*\footnotelang@lua[1] [1=L,usedefault]{%
1927   \ifstrequal{#1}{L}{%
1928     \xright@appenditem{{\csxdef{footnote@luatextextdir}{\the\luatextextdir}}}\to\inserts@list
1929     \global\advance\insert@count \@ne%
1930     \xright@appenditem{{\csxdef{footnote@luatexpardir}{\the\luatexpardir}}}\to\inserts@list
1931     \global\advance\insert@count \@ne%
1932   }%
1933   {%
1934     \xright@appenditem{{\csxdef{footnote@luatextextdir}{\the\luatextextdir}}}\to\inserts@list
1935     \global\advance\insert@countR \@ne%
1936     \xright@appenditem{{\csxdef{footnote@luatexpardir}{\the\luatexpardir}}}\to\inserts@list
1937     \global\advance\insert@countR \@ne%
1938   }%
1939 }
```

`\footnotelang@poly` `\footnotelang@poly` is called to remember the information about the language of a lemma when polyglossia is used.

```

1940 \newcommandx*\footnotelang@poly[1] [1=L,usedefault]{%
1941   \ifstrequal{#1}{L}{%
1942     \if@RTL%
1943       \xright@appenditem{{\csxdef{footnote@dir}{@RTLtrue}}}\to\inserts@list%Know the langua
1944       \global\advance\insert@count \@ne%
1945     \else
1946       \xright@appenditem{{\csxdef{footnote@dir}{@RTLfalse}}}\to\inserts@list%Know the lang
1947       \global\advance\insert@count \@ne%
1948     \fi%
1949     \xright@appenditem{{\csxdef{footnote@lang}{\expandonce\language}}}\to\inserts@list%
```



```

1950 \global\advance\insert@count \@ne%
1951 }%
1952 {%
1953 \ifRTL
1954 \xright@appenditem{{\csxdef{footnote@dir}{@RTLtrue}}}\to\inserts@listR%Know the language of lemma
1955 \global\advance\insert@countR \@ne%
1956 \else
1957 \xright@appenditem{{\csxdef{footnote@dir}{@RTLfalse}}}\to\inserts@listR%Know the language of lemma
1958 \global\advance\insert@countR \@ne%
1959 \fi
1960 \xright@appenditem{{\csxdef{footnote@lang}{\expandonce\language\language}}}\to\inserts@listR%Know the language of lemma
1961 \global\advance\insert@countR \@ne%
1962 }%
1963 }

```

## XII.4 General survey of the way we manage notes

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 are 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`.

These macros are changed depending of the footnotes arrangement: “normal”, “paragraphed”, “two columns” or “three columns”.

## XII.5 General setup

`\footsplitskips` 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 `eledpar`, in order to allow notes to flow from left to right pages and *vice-versa*. Otherwise, we let it to `\@MM`, which is the standard  $\TeX$  `\floatingpenalty`.

```

1964 \newcommand*{\footsplitskips}{%
1965   \interlinepenalty=\interfootnotelinepenalty
1966   \unless\ifl@dprintingpages%
1967     \floatingpenalty=\@MM%
1968   \fi%
1969   \splittopskip=\ht\strutbox \splitmaxdepth=\dp\strutbox
1970   \leftskip=\z@skip \rightskip=\z@skip}
1971

```

`\normalfootnoterule` `\normalfootnoterule` is a standard footnote-rule macro, for use by a `footstart` macro: just the same as the PLAIN T<sub>E</sub>X footnote rule.

```

1972 \let\normalfootnoterule=\footnoterule

```

## XII.6 Footnotes arrangement

### XII.6.1 User level macro

`\Xarrangement` `\Xarrangement[⟨s⟩]{⟨arrangement⟩}` The command calls, for each series, a specific command which set many counters and commands in order to define specific arrangement.

```

1973 \newcommandx{\Xarrangement}[2][1,usedefault]{%
1974   \def\do##1{%
1975     \csname Xarrangement@#2\endcsname{##1}%
1976   }%
1977   \ifstrempy{#1}%
1978     {%
1979       \dolistloop{\@series}%
1980     }%
1981   {
1982     \docsvlist{#1}%
1983   }%
1984 }%

```

### XII.6.2 Normal footnote

`\Xarrangement@normal` We can now define all the parameters for the series of footnotes; initially they use the “normal” footnote formatting.

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 `reledmac` code.)

```

\skip\Afootins=12pt plus5pt minus5pt
\count\Afootins=1000
\dimen\Afootins=0.8\vsizel
\let\vAfootnote=\normalvfootnote \let\Afootfmt=\normalfootfmt
\let\Afootstart=\normalfootstart \let\Afootgroup=\normalfootgroup
\let\Afootnoterule=\normalfootnoterule

```

(Read *The TeXbook* in order to understand what are the counter, skip and dimen associated to an insertion.)

Instead of repeating ourselves, we define a `\Xarrangement@normal` macro that makes all these assignments for us, for any given series letter. This command is called when people use `\Xarrangement[<series>]{normal}`

Now we set up the `\Xarrangement@normal` macro itself. It takes one argument: the footnote series letter.

```

1985 \newcommand*{\Xarrangement@normal}[1]{%
1986   \csgdef{series@display#1}{normal}
1987   \expandafter\let\csname #1footstart\endcsname=\normalfootstart
1988   \expandafter\let\csname v#1footnote\endcsname=\normalvfootnote
1989   \expandafter\let\csname #1footfmt\endcsname=\normalfootfmt
1990   \expandafter\let\csname #1footgroup\endcsname=\normalfootgroup
1991   \expandafter\let\csname #1footnoterule\endcsname=%
1992                                   \normalfootnoterule
1993   \count\csname #1footins\endcsname=1000
1994   \dimen\csname #1footins\endcsname=\csuse{Xmaxhnotes@#1}
1995   \skip\csname #1footins\endcsname=\csuse{Xbeforenotes@#1}%
1996   \advance\skip\csname #1footins\endcsname by\csuse{Xafterrule@#1}%

```

The `reledpar` provides tools in order to confine notes to one side. The mechanism is explained in the `reledpar`'s handbook. For now, just retain we need to store default value of the counter associated to the notes  $\TeX$ 's inserts.

```

1997 \csxdef{default@#1footins}{1000}%Use this to confine the notes to one side only

```

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).

```

1998 \ifnoledgroup@else%
1999   \expandafter\let\csname mpv#1footnote\endcsname=\mpnormalvfootnote
2000   \expandafter\let\csname mp#1footgroup\endcsname=\mpnormalfootgroup
2001   \count\csname mp#1footins\endcsname=1000
2002   \dimen\csname mp#1footins\endcsname=\csuse{Xmaxhnotes@#1}
2003   \skip\csname mp#1footins\endcsname=\csuse{Xbeforenotes@#1}%
2004   \advance\skip\csname mp#1footins\endcsname by\csuse{Xafterrule@#1}%
2005 \fi
2006 }
2007

```

`\normalvfootnote` 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.

```

2008 \notbool{parapparatus@}{\newcommand*}{\newcommand}{\normalvfootnote}[2]{%
2009   \insert\csname #1footins\endcsname\bgroup
2010   \csuse{Xhooknote@#1}
2011   \csuse{Xnotefontsize@#1}
2012   \footplitskips
2013   \ifl@dpairing\ifl@dpaging\else%
2014     \setXnoteswidthliketwocolumns@{#1}%
2015   \fi\fi%

```

```

2016 \setXnotespositionliketwocolumns@{#1}%
2017 \spaceskip=\z@skip \xspaceskip=\z@skip
2018 \csname #1footfmt\endcsname #2{#1}\egroup

```

`\mpnormalvfootnote` And a somewhat different version for minipages.

```

2019 \notbool{parapparatus@}{\newcommand*}{\newcommand}{\mpnormalvfootnote}[2]{%
2020 \global\setbox\@nameuse{mp#1footins}\vbox{%
2021 \unvbox\@nameuse{mp#1footins}
2022 \csuse{Xhooknote@#1}
2023 \csuse{Xnotefontsize@#1}
2024 \hsize\columnwidth
2025 \@parboxrestore
2026 \color@begingroup
2027 \csname #1footfmt\endcsname #2{#1}\color@endgroup}}
2028

```

`\normalfootfmt` `\normalfootfmt` is a ‘normal’ macro to take the footnote line and page number information (see V.9 p. 77), and the desired text, and output what’s to be printed. Argument #1 contains the line and page number information and lemma font specifier; #2 is the lemma; #3 is the note’s text. This version is very rudimentary—it uses `\printlines` to print just the range of line numbers, followed by a square bracket, the lemma, and the note text.

```

2029
2030
2031 \notbool{parapparatus@}{\newcommand*}{\newcommand}{\normalfootfmt}[4]{%
2032 \Xledsetnormalparstuff{#4}%
2033 \hangindent=\csuse{Xhangindent@#4}
2034 \strut{\printlinefootnote{#1}{#4}}%
2035 {\nottoggle{Xlemmadisablefontselection@#4}{\select@lemmafont#1|#2}{#2}}%
2036 \iftoggle{nosep@}{\hskip\csuse{Xinplaceoflemmaseparator@#4}}{\ifcsemt{Xlemmaseparator@
2037 {\hskip\csuse{Xinplaceoflemmaseparator@#4}}%
2038 {\nobreak\hskip\csuse{Xbeforelemmaseparator@#4}\csuse{Xlemmaseparator@#4}\hskip\csuse{X
2039 }}%
2040 #3\strut\par}

```

`\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\Xfootins` 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 is used instead of `\skip\footins` for the first printed series in one page.

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 `reledmac` in this

respect is general across all footnote types. 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`.

2041 `\newcommand*{\normalfootstart}[1]{%`

The first series of notes printed in a page can have a specific skip before it. In order to insert this specific skip without overlap the bottom margin of the page, Maïeul Rouquette have defined an algorithm explained in XVIII p. 169. Here is part of this algorithm, when the block of notes are ready to be printed.

```
2042   \ifdimequal{0pt}{\preXnotes@}{}%
2043   {%
2044     \iftoggle{preXnotes@}{%
2045       \togglefalse{preXnotes@}%
2046       \skip\csname #1footins\endcsname=%
2047       \dimexpr\csuse{preXnotes@}+\csuse{Xafterrule@#1}\relax%
2048     }%
2049   }%
2050 }%
2051 \vskip\skip\csname #1footins\endcsname%
```

And now, the problem of left and right skip for notes. Especially when using one feature of `reledpar` which allows to have the footnotes horizontal size as the size of columns printed by `\Columns`. Read XV p. 167 for the general description of the problem.

```
2052 \leftskip0pt \rightskip0pt
2053 \ifl@dpairing\else%
2054   \hsize=\old@hsize%
2055 \fi%
2056 \setXnoteswidthliketwocolumns@{#1}%
2057 \setXnotespositionliketwocolumns@{#1}%
2058 % \end{macrocode}
2059 % And now, print the footnote's rule to finish the footnote's introduction.
2060 % \begin{macrocode}
2061 \print@Xfootnoterule{#1}%
2062 \noindent\leavevmode}
```

`\normalfootgroup` `\normalfootgroup` is a standard footnote-grouping macro: it sends the contents of the footnote-insert box to the output page without alteration.

```
2063 \newcommand*{\normalfootgroup}[1]{%
2064   {\csuse{Xnotefontsize@#1}\noindent\csuse{Xtxtbeforenotes@#1}}%
2065   \unvbox\csname #1footins\endcsname%
2066   \hsize=\old@hsize%
2067 }%
2068
```

`\mpnormalfootgroup` A somewhat different version for minipages. Notes that, in this case, we don not make distinction between `\Xfootgroup` and `\Xfootstarts` macro.

```
2069 \unless\ifnoledgroup@
2070 \newcommand*{\mpnormalfootgroup}[1]{%
2071   \vskip\skip\@nameuse{mp#1footins}
```

```

2072 \ifl@dpairing\ifparledgroup%
2073   \leavevmode\marks\parledgroup@{begin}%
2074   \marks\parledgroup@series{#1}%
2075   \marks\parledgroup@type{Xfootnote}%
2076 \fi\fi\normalcolor%
2077 \ifparledgroup%
2078   \ifl@dpairing%
2079   \else%
2080     \setXnoteswidthliketwocolumns@{#1}%
2081     \setXnotespositionliketwocolumns@{#1}%
2082     \print@Xfootnoterule{#1}%%
2083   \fi%
2084 \else%
2085   \setXnoteswidthliketwocolumns@{#1}%
2086   \setXnotespositionliketwocolumns@{#1}%
2087   \print@Xfootnoterule{#1}%%
2088 \fi%
2089 \setlength{\parindent}{0pt}
2090 {\csuse{Xnotefontsize@#1}\csuse{Xtxtbeforenotes@#1}}
2091 \unvbox\csname mp#1footins\endcsname}}
2092 \fi

```

### XII.6.3 Paraphrased footnotes

The paraphrased-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 *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.

`\Xarrangement@paragraph` The `\Xarrangement@paragraph` macro sets up everything for one series of the footnotes so that they will be paraphrased; it takes the series letter as argument. We include the setting of `\count\footins` to 1000 for the footnote series just in case user is switching to paraphrased footnotes after having columnar ones, since they change this value (see below).

The argument of `\Xarrangement@footparagraph` is the letter denoting the series of notes to be paraphrased.

```

2093 \newcommand*{\Xarrangement@paragraph}[1]{%
2094   \csgdef{series@display#1}{paragraph}
2095   \expandafter\newcount\csname #1prevpage@num\endcsname
2096   \expandafter\let\csname #1footstart\endcsname=\parafootstart
2097   \expandafter\let\csname v#1footnote\endcsname=\paravfootnote
2098   \expandafter\let\csname #1footfmt\endcsname=\parafootfmt
2099   \expandafter\let\csname #1footgroup\endcsname=\parafootgroup
2100   \count\csname #1footins\endcsname=1000
2101   \csxdef{default@#1footins}{1000}%Use this to confine the notes to one side only
2102   \dimen\csname #1footins\endcsname=\csuse{Xmaxhnotes@#1}
2103   \skip\csname #1footins\endcsname=\csuse{Xbeforenotes@#1}%
2104   \advance\skip\csname #1footins\endcsname by\csuse{Xafterterrule@#1}%

```

```

2105 \para@footsetup{#1}
      And the extra setup for minipages.
2106 \ifnoledgroup@else
2107   \expandafter\let\csname mpv#1footnote\endcsname=\mpparavfootnote
2108   \expandafter\let\csname mp#1footgroup\endcsname=\mpparafootgroup
2109   \count\csname mp#1footins\endcsname=1000
2110   \dimen\csname mp#1footins\endcsname=\csuse{Xmaxhnotes@#1}
2111   \skip\csname mp#1footins\endcsname=\csuse{Xbeforenotes@#1}%
2112   \advance\skip\csname mp#1footins\endcsname by\csuse{Xafterrule@#1}%
2113 \fi
2114 }

```

`\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.

```
2115 \providecommand{\footfudgefiddle}{64}
```

`\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 9 pt) has been set already. The argument of the macro is again the note series letter.

Peter Wilson thinks that `\columnwidth` should be used here for  $\LaTeX$  not `\hsize`. Peter Wilson have also included `\footfudgefiddle`.

```

2116 \newcommand*{\para@footsetup}[1]{\csuse{Xnotefontsize@#1}
2117   \setXnoteswidthliketwocolumns@{#1}%
2118   \dimen0=\baselineskip
2119   \multiply\dimen0 by 1024
2120   \divide\dimen0 by\columnwidth\multiply\dimen0 by\footfudgefiddle\relax
2121   \csxdef{#1footfudgefactor}{%
2122     \expandafter\strip@pt\dimen0 }}
2123

```

`\strip@pt` strip the characters pt from a dimen value.

`\parafootstart` `\parafootstart` is the same as `\normalfootstart`, 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). 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.

```

2124 \newcommand*{\parafootstart}[1]{%
2125   \rightskip=0pt \leftskip=0pt \parindent=0pt
2126   \ifdimequal{0pt}{\preXnotes@}{}%
2127   {%
2128     \iftoggle{preXnotes@}{%
2129       \togglefalse{preXnotes@}%
2130       \skip\csname #1footins\endcsname=%

```

```

2131          \dimexpr\csuse{preXnotes@}+\csuse{Xafterrule@#1}\relax%
2132      }%
2133  {}%
2134  }%
2135  \vskip\skip\csname #1footins\endcsname%
2136  \setXnoteswidthliketwocolumns@{#1}%
2137  \setXnotespositionliketwocolumns@{#1}%
2138  \print@Xfootnoterule{#1}%
2139  \noindent\leavevmode}

```

`\paravfootnote` `\paravfootnote` is a version of the `\vfootnote` command that is used for paragraphed notes. It gets appended to the `\inserts@list` list by an outer-level footnote command like `\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 `\insert\footins` definition in *The TeXbook*, 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 `\discretionary`s. If you later unbox these `hboxes` and stick them together, as the *TeXbook* macros do to make these footnotes, you lose the ability to hyphenate after an explicit hyphen. This can lead to overfull `hboxes` when you would not expect to find them, and to the uninitiated it might be very hard to see why the problem had arisen.<sup>27</sup>

Wayne Sullivan pointed out to us another subtle problem that arises from the same cause:  $\TeX$  also leaves the `\language` whatsit nodes out of the horizontal list.<sup>28</sup> 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.

To get around these problems, Wayne suggested emendations to the *TeXbook* 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 `hbox` in the first place, but instead to collect it in a `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 `vbox`, as well as the `hboxes` inside it, but that is not too hard. For details, we refer you to Michael's article, where the issues are clearly explained.<sup>29</sup> Michael's unboxing macro is called `\Xunvxh`: `unvbox`, 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 `vbox` the way we are doing.<sup>30</sup>

<sup>27</sup>Michael Downes, 'Line Breaking in `\unhboxed` Text', *TUGboat* **11** (1990), pp. 605–612.

<sup>28</sup>See *The TeXbook*, p. 455 (editions after January 1990).

<sup>29</sup>Wayne supplied his own macros to do this, but since they were almost identical to Michael's, Peter Wilson have used the latter's `\Xunvxh` macro since it is publicly documented.

<sup>30</sup>'Line Breaking', p. 610.



In other words, be very careful not to say `\break`, or `\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 `\penalty-9999` will be quite okay. Just do not make the break mandatory. We have not applied any of Michael's solutions here, since we feel that the problem is exiguous, and `reledmac` is quite baroque enough already. If you think you are having this problem, look up Michael's solutions.

One more thing; we set `\leftskip` and `\rightskip` to zero. This has the effect of neutralizing any such skips which may apply to the main text (cf. XII.6.2 p. 132 above). We need to do this, since `\footfudgefactor` is calculated on the assumption that the notes are `\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.

```

2140 \newcommand*{\paravfootnote}[2]{%
2141   \insert\csname #1footins\endcsname
2142   \bgroup
2143     \csuse{Xhooknote@#1}
2144     \csuse{Xnotefontsize@#1}
2145     \footplitskips
2146     \setbox0=\vbox{\hsize=\maxdimen
2147       \noindent\csname #1footfmt\endcsname #2{#1}}%
2148     \setbox0=\hbox{\Xunvxh{0}{#1}}%
2149     \dp0=0pt
2150     \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.

```

2151   \if@RTL\noindent \leavevmode\fi\box0%
2152   \penalty0
2153   \egroup}
2154

```

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 *The TeXbook*, p. 124),  $\TeX$  inserts a penalty of  $-10000$  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 does not force, such breaks. This penalty of 0 is later removed, after page breaks have been decided, by the `\unpenalty` macro in `\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`).

`\mpparavfootnote` This version is for minipages.

```

2155 \newcommand*{\mpparavfootnote}[2]{%
2156   \global\setbox\@nameuse{mp#1footins}\vbox{%
2157     \unvbox\@nameuse{mp#1footins}%
2158     \csuse{Xhooknote@#1}
2159     \csuse{Xnotefontsize@#1}

```

```

2160 \footsplitskips
2161 \setbox0=\vbox{\hsize=\maxdimen
2162 \noindent\color@begingroup\csname #1footfmt\endcsname #2{#1}\color@endgroup}%
2163 \setbox0=\hbox{\Xunvxh{0}{#1}}%
2164 \dp0=\z@
2165 \ht0=\csname #1footfudgefactor\endcsname\wd0
2166 \box0
2167 \penalty0
2168 }}
2169

```

`\Xunvxh` 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 TeXbook*, pp. 99–100). `\unvxh` cancels these unwanted paragraph-final items using `\unskip` and `\unpenalty`.

```

2170 \newcommand*{\Xunvxh}[2]{%
2171   \setbox0=\vbox{\unvbox#1%
2172     \global\setbox1=\lastbox}%
2173   \unhbox1
2174   \unskip           % remove \rightskip,
2175   \unskip           % remove \parfillskip,
2176   \unpenalty        % remove \penalty of 10000,
2177   \hskip\csuse{Xafternote@#2}} % but add the glue to go between the notes
2178

```

`\parafootfmt` `\parafootfmt` is `\normalfootfmt` adapted to do the special stuff needed for paragraphed 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).

```

2179 \newcommand*{\parafootfmt}[4]{%
2180   \Xinsertparafootsep{#4}%
2181   \Xledsetnormalparstuff{#4}%
2182   \printlinefootnote{#1}{#4}%
2183   {\nottoggle{Xlemmadisablefontselection@#4}{\select@lemmafont#1|#2}{#2}}%
2184   \iftoggle{nosep@}{\hskip\csuse{Xinplaceoflemmaseparator@#4}}{\ifcsemtyp{Xlemmaseparator@
2185     {\hskip\csuse{Xinplaceoflemmaseparator@#4}}%
2186     {\nobreak\hskip\csuse{Xbeforelemmaseparator@#4}\csuse{Xlemmaseparator@#4}\hskip\csuse{X
2187   }}%
2188   #3\penalty-10 }

```

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 `\Xinsertparafootsep` command is used to insert the `\Xparafootsep@series` between each note in the *same* page.

`\parafootgroup` This footgroup code is modelled on the macros in *The TeXbook*, p. 399. The only difference is the `\unpenalty` in `\makehboxofhboxes`, which is there to remove the penalty

of 0 which was added to the end of each footnote by `\para@vfootnote`.

The call to `\Xnotefontsize@<s>` is to ensure that the correct `\baselineskip` for the footnotes is used. The argument is the note series letter.

```

2189 \newcommand*{\parafootgroup}[1]{%
2190   \unvbox\csname #1footins\endcsname
2191   \ifcsstring{Xragged@#1}{L}{\RaggedLeft}{}%
2192   \ifcsstring{Xragged@#1}{R}{\RaggedRight}{}%
2193   \makehboxofhboxes
2194   \setbox0=\hbox{\csuse{Xnotefontsize@#1}\csuse{Xtxtbeforenotes@#1}}\unhbox0 \removehboxes}%
2195   \csuse{Xnotefontsize@#1}
2196   \noindent\unhbox0\par%
2197   \global\hsize=\old@hsize%
2198   }%
2199

```

`\mpparafootgroup` The minipage version.

```

2200 \newcommand*{\mpparafootgroup}[1]{%
2201   \setXnoteswidthliketwocolumns@{#1}%
2202   \vskip\skip\@nameuse{mp#1footins}
2203   \ifl@dpairing\ifparledgroup%
2204     \leavevmode\marks\parledgroup@{begin}%
2205     \marks\parledgroup@series{#1}%
2206     \marks\parledgroup@type{Xfootnote}%
2207     \fi\fi\normalcolor
2208     \ifparledgroup%
2209       \ifl@dpairing%
2210         \else%
2211           \setXnoteswidthliketwocolumns@{#1}%
2212           \setXnotespositionliketwocolumns@{#1}%
2213           \print@Xfootnoterule{#1}%
2214         \fi%
2215       \else%
2216         \setXnoteswidthliketwocolumns@{#1}%
2217         \setXnotespositionliketwocolumns@{#1}%
2218         \print@Xfootnoterule{#1}%
2219       \fi%
2220     \unvbox\csname mp#1footins\endcsname
2221     \ifcsstring{Xragged@#1}{L}{\RaggedLeft}{}%
2222     \ifcsstring{Xragged@#1}{R}{\RaggedRight}{}%
2223     \makehboxofhboxes
2224     \setbox0=\hbox{\csuse{Xnotefontsize@#1}\csuse{Xtxtbeforenotes@#1}}\unhbox0 \removehboxes}%
2225     \csuse{Xnotefontsize@#1}
2226     \noindent\unhbox0\par}}
2227

```

And finally, the two macros which are required to transform the long horizontal box stored in the insert' box to a printable text.

```

\makehboxofhboxes
\removehboxes

```

```

2228 \newcommand*{\makehboxofhboxes}{\setbox0=\hbox{}}%
2229 \loop
2230   \unpenalty
2231   \setbox2=\lastbox
2232   \ifhbox2
2233     \setbox0=\hbox{\box2\unhbox0}%
2234   \repeat}
2235
2236 \newcommand*{\removehboxes}{\setbox0=\lastbox
2237   \ifhbox0{\removehboxes}\unhbox0 \fi}
2238

```

**Insertion of the footnotes separator** The command `\Xinsertparafootsep{<series>}` must be called at the beginning of `\parafootftm`.

```

\prevpage@num
\Xinsertparafootsep 2239 \newcommand{\Xinsertparafootsep}[1]{%
2240   \ifnumequal{csuse{#1prevpage@num}}{\page@num}%
2241     {\ifcsdef{prevline#1}% Be sur \prevline#1 exists.
2242       {\ifnumequal{csuse{prevline#1}}{\line@num}%
2243         {\ifcempty{Xsymlinenum@#1}{\csuse{Xparafootsep@#1}}{}}}%
2244       {\csuse{Xparafootsep@#1}}}%
2245   }%
2246   {\csuse{Xparafootsep@#1}}}%
2247 }%
2248 {}%
2249 \global\csname #1prevpage@num\endcsname=\page@num%
2250 }

```

## XII.6.4 Columnar footnotes

### Common tools

`\rigidbalance` 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 into a number (#2) of columns, each with a space (#3) between the top baseline and the top of the `\vbox`. The `\splitoff` macro is taken from *The TeXbook*, p. 397, with a slight change to the syntax of the arguments so that they do not 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  $\text{\LaTeX}$  `\line` macro has no relationship to the TeX `\line`. The  $\text{\LaTeX}$  equivalent is `\@@line`.

```

2251 \newcount\@k \newdimen\@h
2252 \newcommand*{\rigidbalance}[3]{\setbox0=\box#1 \@k=#2 \@h=#3
2253   \@@line{\splittopskip=\@h \vbadness=\@M \hfilneg
2254     \valign{##\vfil\cr\dosplits}}}%
2255
2256 \newcommand*{\dosplits}{\ifnum\@k>0 \noalign{\hfil}\splitoff
2257   \global\advance\@k-1\cr\dosplits\fi}

```

```

2258
2259 \newcommand*{\splitoff}{\dimen0=\ht0
2260   \divide\dimen0 by\@k \advance\dimen0 by\@h
2261   \setbox2 \vsplit0 to \dimen0
2262   \unvbox2 }
2263

```

### Three columns

`\Xarrangement@threecol`

```

2264 \newcommand*{\Xarrangement@threecol}[1]{%
2265   \csgdef{series@display#1}{threecol}
2266   \expandafter\let\csname v#1footnote\endcsname=\threecolvfootnote
2267   \expandafter\let\csname #1footfmt\endcsname=\threecolfootfmt
2268   \expandafter\let\csname #1footgroup\endcsname=\threecolfootgroup
2269   \dimen\csname #1footins\endcsname=\csuse{Xmaxhnotes@#1}%
2270   \skip\csname #1footins\endcsname=\csuse{Xbeforenotes@#1}%
2271   \advance\skip\csname #1footins\endcsname by\csuse{Xafterrule@#1}%
2272   \threecolfootsetup{#1}

```

The additional setup for minipages.

```

2273 \ifnoledgroup@else
2274   \expandafter\let\csname mpv#1footnote\endcsname=\mpnormalvfootnote
2275   \expandafter\let\csname mp#1footgroup\endcsname=\mpthreecolfootgroup
2276   \skip\csname mp#1footins\endcsname=\csuse{Xbeforenotes@#1}%
2277   \advance\skip\csname mp#1footins\endcsname by\csuse{Xafterrule@#1}%
2278   \mpthreecolfootsetup{#1}
2279 \fi
2280 }
2281

```

The `\footstart` and `\footnoterule` macros for these notes assume the normal values (XII.6.2 p. 132 above).

`\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 `\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 does not apply the `\count` scaling.

```

2282 \newcommand*{\threecolfootsetup}[1]{%
2283   \count\csname #1footins\endcsname 333

```

```

2284 \csxdef{default@#1footins}{333}%Use this to confine the notes to one side only
2285 \multiply\dimen\csname #1footins\endcsname \thr@@}

\mpthreecolfootsetup The setup for minipages.
2286 \newcommand*{\mpthreecolfootsetup}[1]{%
2287 \count\csname mp#1footins\endcsname 333
2288 \multiply\dimen\csname mp#1footins\endcsname \thr@@}
2289

\threecolvffootnote \threecolvffootnote is the \vfootnote command for three-column notes. The call
to \Xnotefontsize@<s> ensures that the \splittopskip and \splitmaxdepth take
their values from the right \strutbox: the one used in a footnotes. Note especially
the importance of temporarily reducing the \hsize to 0.3 of its normal value. This
determines the widths of the individual columns. So if the normal \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 #1 the full text of the note (including
numbers, lemma and text).
2290 \notbool{parapparatus@}{\newcommand*}{\newcommand}{\threecolvffootnote}[2]{%
2291 \insert\csname #1footins\endcsname\bgroup
2292 \csuse{Xnotefontsize@#1}
2293 \footsplitskips
2294 \csname #1footfmt\endcsname #2{#1}\egroup}

\threecolfootfmt \threecolfootfmt is the command that formats one note. The arguments are #1 the
line numbers, #2 the lemma and #4 the text of the -footnote command #4 optional
(for backward compatibility): the series.
2295 \notbool{parapparatus@}{\newcommand*}{\newcommand}{\threecolfootfmt}[4]{%
2296 \normal@pars
2297 \hsize \csuse{Xsizethreecol@#4}
2298 \nottoggle{Xparindent@#4}{\parindent=\z@}{%
2299 \tolerance=5000
2300 \hangindent=\csuse{Xhangindent@#4}
2301 \leavevmode
2302 \csuse{Xcolalign@#4}%
2303 \strut{\printlinefootnote{#1}{#4}}%
2304 {\nottoggle{Xlemmadisablefontselection@#4}{\select@lemfont#1|#2}{#2}}%
2305 \iftoggle{nosep@}{\hskip\csuse{Xinplaceoflemmaseparator@#4}}{\ifcsemt{Xlemmaseparator@#4}}%
2306 {\hskip\csuse{Xinplaceoflemmaseparator@#4}}%
2307 {\nobreak\hskip\csuse{Xbeforelemmaseparator@#4}\csuse{Xlemmaseparator@#4}\hskip\csuse{X
2308 }}%
2309 #3\strut\par\allowbreak}

\threecolfootgroup And here is the footgroup macro that is called within the output routine to regroup
the notes into three columns. Once again, the call to \Xnotefontsize@<s> 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.

```
2310 \newcommand*{\threecolfootgroup}[1]{\csuse{Xnotefontsize@#1}%
2311 \noindent\csuse{Xtxtbeforenotes@#1}\par%
2312 \splittopskip=\ht\strutbox
2313 \expandafter
2314 \rigidbalance\csname #1footins\endcsname \thr@@ \splittopskip}}
```

`\mpthreecolfootgroup` The setup for minipages.

```
2315 \newcommand*{\mpthreecolfootgroup}[1]{%
2316 \vskip\skip\@nameuse{mp#1footins}
2317 \ifl@dpairing\ifparledgroup%
2318 \leavevmode\marks\parledgroup@{begin}%
2319 \marks\parledgroup@series{#1}%
2320 \marks\parledgroup@type{Xfootnote}%
2321 \fi\fi\normalcolor
2322 \ifparledgroup%
2323 \ifl@dpairing%
2324 \else%
2325 \setXnoteswidthliketwocolumns@{#1}%
2326 \setXnotespositionliketwocolumns@{#1}%
2327 \print@Xfootnoterule{#1}%
2328 \fi%
2329 \else%
2330 \setXnoteswidthliketwocolumns@{#1}%
2331 \setXnotespositionliketwocolumns@{#1}%
2332 \print@Xfootnoterule{#1}%
2333 \fi%
2334 {\csuse{Xnotefontsize@#1}\noindent\csuse{Xtxtbeforenotes@#1}\par
2335 \splittopskip=\ht\strutbox
2336 \expandafter
2337 \rigidbalance\csname mp#1footins\endcsname \thr@@ \splittopskip}}
2338
```

## Two columns

`\Xarrangement@twocol`

```
2339 \newcommand*{\Xarrangement@twocol}[1]{%
2340 \csgdef{series@display#1}{twocol}
2341 \expandafter\let\csname v#1footnote\endcsname=\twocolvfootnote
2342 \expandafter\let\csname #1footfmt\endcsname=\twocolfootfmt
2343 \expandafter\let\csname #1footgroup\endcsname=\twocolfootgroup
2344 \dimen\csname #1footins\endcsname=\csuse{Xmaxhnotes@#1}%
2345 \skip\csname #1footins\endcsname=\csuse{Xbeforenotes@#1}%
2346 \advance\skip\csname #1footins\endcsname by\csuse{Xafterterrule@#1}%
2347 \twocolfootsetup{#1}}
```

The additional setup for minipages.

```

2348 \ifnoledgroup@ \else
2349   \expandafter \let \csname mpv#1footnote \endcsname = \mpnormalvfootnote
2350   \expandafter \let \csname mp#1footgroup \endcsname = \mptwocolfootgroup
2351   \skip \csname mp#1footins \endcsname = \csuse{Xbeforenotes@#1}%
2352   \advance \skip \csname mp#1footins \endcsname by \csuse{Xafterrule@#1}%
2353   \mptwocolfootsetup{#1}
2354 \fi
2355 }
2356
\twocolfootsetup Here is a series of macros which are very similar to their three-column counterparts. In
\twocolvfootnote this case, each note is assumed to contribute only a half a line of text. And the notes are
\twocolfootfmt set in columns giving a gap between them of one tenth of the \hszize.
\twocolfootgroup
2357 \newcommand*{\twocolfootsetup}[1]{%
2358   \count \csname #1footins \endcsname 500
2359   \csxdef{default@#1footins}{500}% Use this to confine the notes to one side only
2360   \multiply \dimen \csname #1footins \endcsname \tw@
2361   \notbool{parapparatus@}{\newcommand*{\newcommand}{\twocolvfootnote}[2]{\insert \csname #1
2362     \csuse{Xnotefontsize@#1}
2363     \footsplitskips
2364     \csname #1footfmt \endcsname #2{#1}\egroup}
2365   \notbool{parapparatus@}{\newcommand*{\newcommand}{\twocolfootfmt}[4]{% 4th arg is optional
2366     \normal@pars
2367     \hszize \csuse{Xhsizetwocol@#4}
2368     \nottoggle{Xparindent@#4}{\parindent=\z@}{%
2369       \tolerance=5000
2370       \hangindent=\csuse{Xhangindent@#4}
2371       \leavevmode
2372       \csuse{Xcolalign@#4}%
2373       \strut{\printlinefootnote{#1}{#4}}%
2374       {\nottoggle{Xlemmadisablefontselection@#4}{\select@lemmafnt#1|#2}{#2}}%
2375       \iftoggle{nosep@}{\hskip \csuse{Xinplaceoflemmaseparator@#4}}{\ifcsemtty{Xlemmaseparator@
2376         {\hskip \csuse{Xinplaceoflemmaseparator@#4}}%
2377         {\nobreak \hskip \csuse{Xbeforelemmaseparator@#4} \csuse{Xlemmaseparator@#4} \hskip \csuse{X
2378         }}%
2379       #3 \strut \par \allowbreak}
2380   \newcommand*{\twocolfootgroup}[1]{\csuse{Xnotefontsize@#1}
2381     \noindent \csuse{Xtxtbeforenotes@#1} \par%
2382     \splittopskip=\ht \strutbox
2383     \expandafter
2384     \rigidbalance \csname #1footins \endcsname \tw@ \splittopskip}
2385
\mptwocolfootsetup The versions for minipages.
\mptwocolfootgroup
2386 \newcommand*{\mptwocolfootsetup}[1]{%
2387   \count \csname mp#1footins \endcsname 500
2388   \multiply \dimen \csname mp#1footins \endcsname \tw@

```



```

2389 \newcommand*{\mptwocolfootgroup}[1]{%
2390   \vskip\skip\@nameuse{mp#1footins}
2391   \ifl@dpairing\ifparledgroup%
2392     \leavevmode\marks\parledgroup@{begin}%
2393     \marks\parledgroup@series{#1}%
2394     \marks\parledgroup@type{Xfootnote}%
2395   \fi\fi\normalcolor
2396   \ifparledgroup%
2397     \ifl@dpairing%
2398     \else%
2399       \setXnoteswidthliketwocolumns@{#1}%
2400       \setXnotespositionliketwocolumns@{#1}%
2401       \print@Xfootnoterule{#1}%
2402     \fi%
2403   \else%
2404     \setXnoteswidthliketwocolumns@{#1}%
2405     \setXnotespositionliketwocolumns@{#1}%
2406     \print@Xfootnoterule{#1}%
2407   \fi%
2408   {\csuse{Xnotefontsize@#1}\noindent\csuse{Xtxtbeforenotes@#1}}\par
2409   \splittopskip=\ht\strutbox
2410   \expandafter
2411   \rigidbalance\csname mp#1footins\endcsname \tw@ \splittopskip}}
2412

```

## XII.7 Critical notes presentation

Here, we define some commons macro which are used in order to print a critical notes, that is a note with 1) line number 2) lemma 3) lemma separator 4) text associated to the lemma.

### XII.7.1 Font tools

`\endashchar` 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.

```

2413 \def\endashchar{\textnormal{--}}
2414 \newcommand*{\fullstop}{\textnormal{.}}

```

```

2415 \newcommand*{\rbracket}{\textnormal{%
2416   \csuse{text\csuse{footnote@lang}}{%
2417     \ifluatex%
2418       \ifdefstring{\footnote@luatextextdir}{TRT}{\thinspace[]\thinspace}}%
2419     \else%
2420       \thinspace}%
2421     \fi}%
2422 }%
2423 }
2424

```

### XII.7.2 Pstart number in footnote

`\printpstart` The `\printpstart` macro prints the pstart number for a note.

```

2425 \newcommand{\printpstart}[0]{%
2426   \ifboolexpr{bool{!@dpairing} or bool{!@dprintingpages} or bool{!@dprintingcolumns}}{%
2427     \ifledRcol%
2428       \thepstartR%
2429     \else%
2430       \thepstartL%
2431     \fi%
2432   }{%
2433     \thepstart%
2434   }%
2435 }

```

### XII.7.3 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 series options. Its first argument is the information about lines; its second is the series of the footnote. The printing of the line number is shared in `\printlinefootnotennumbers`.

```

2436 \newcommand{\printlinefootnote}[2]{%
2437   \def\extractline@##1|##2|##3|##4|##5|##6|##7|{##2}%
2438   \def\extractsubline@##1|##2|##3|##4|##5|##6|##7|{##3}%
2439   \def\extractendline@##1|##2|##3|##4|##5|##6|##7|{##5}%
2440   \def\extractendsubline@##1|##2|##3|##4|##5|##6|##7|{##6}%
2441   \iftoggle{Xnumberonlyfirstintwolines@#2}{%
2442     \edef\lineinfo@{\extractline@ #1| - \extractsubline@ #1| - \extractendline@ #1| - \extractendsubline@ #1}%
2443     }%
2444     {%
2445       \edef\lineinfo@{\extractline@ #1| - \extractsubline@ #1|}%
2446     }%
2447   \iftoggle{nonum@}{%Try if the line number must printed for this specific not (by default, yes)}%
2448   \hspace{\csuse{Xinplaceofnumber@#2}}%
2449   }%
2450   {%
2451     {%
2452       \iftoggle{Xnonumber@#2}%Try if the line number must printed (by default, yes)

```

```

2453      {%
2454      \hspace{\csuse{Xinplaceofnumber@#2}}%
2455      }%
2456      {%
2457      {\iftoggle{Xnumberonlyfirstinline@#2}% If for this series the line number must be printed only i
2458      {%
2459      \ifcsdef{prevline#2}%
2460      {%Be sure the \prevline exists.
2461      \ifcsequal{prevline#2}{\lineinfo@}%Try it
2462      {%
2463      \ifcseempty{Xsymlinenumber@#2}%
2464      {%
2465      \hspace{\csuse{Xinplaceofnumber@#2}}%
2466      }%
2467      {\hspace{\csuse{Xbeforeasymlinenumber@#2}}\csuse{Xnotenumfont@#2}%
2468      \ifdimequal{\csuse{Xboxasymlinenumber@#2}}{0pt}%
2469      {\csuse{Xsymlinenumber@#2}}%
2470      {\hbox to \csuse{Xboxasymlinenumber@#2}{\csuse{Xsymlinenumber@#2}\hfill}}%
2471      \hspace{\csuse{Xafterasymlinenumber@#2}}}%
2472      }%
2473      {%
2474      \printlinefootnotearea{#1}{#2}%
2475      }%
2476      }%
2477      {%
2478      \printlinefootnotearea{#1}{#2}%
2479      }%
2480      }%
2481      {%
2482      \printlinefootnotearea{#1}{#2}%
2483      }%
2484      \csxdef{prevline#2}{\lineinfo@}%
2485      }%
2486      }%
2487      }%
2488      }%
2489  }

```

`\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.)

```

2490 \newcommand{\printlinefootnotearea}[2]{%
2491   \printXbeforenumber{#2}%
2492   \csuse{Xnotenumfont@#2}%
2493   \boxfootnotenumbers{#1}{#2}%
2494   \printXafternumber{#2}%
2495 }%

```

`\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.

```

2496 \newcommand{\boxfootnotenumbers}[2]{%
2497   \ifdimequal{\csuse{Xboxlinenum@#2}}{0pt}{%
2498     \printlinefootnotenumbers{#1}{#2}%
2499   }%
2500   {%
2501     \hbox to \csuse{Xboxlinenum@#2}%
2502     {%
2503       \IfSubStr{RC}{\csuse{Xboxlinenumalign@#2}}{\hfill}{}%
2504       \printlinefootnotenumbers{#1}{#2}%
2505       \IfSubStr{LC}{\csuse{Xboxlinenumalign@#2}}{\hfill}{}%
2506     }%
2507   }%
2508 }%
```

`\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.

```

2509 \newcommand{\printlinefootnotenumbers}[2]{%
2510   \xdef\@currentseries{#2}%
2511   \ifboolexpr{%
2512     (togl{Xpstart@#2} and bool{numberpstart})%
2513     or togl{Xpstarteverytime@#2}}%
2514   {\printpstart}{}%
2515   \iftoggle{Xonlypstart@#2}{\printlines#1|}%
2516 }%
```

`\printXbeforenumber` This macro prints a space (before the line number) in footnote. It is called by `\printlinefootnotearea`. Its only argument is the note series (A, B, C, etc.)

```

2517 \newcommand{\printXbeforenumber}[1]{%
2518   \hspace{\csuse{Xbeforenumber@#1}}%
2519 }%
```

`\printXafternumber` 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

```

2520 \newcommand{\printXafternumber}[1]{%
2521   \iftoggle{Xnonbreakableafternumber@#1}{\nobreak}{}%
2522   \hspace{\csuse{Xafternumber@#1}}%
2523 }%
```

If we have decided to print the line number in a specific notes, 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 V.9 p. 77: 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.

edmac’ creator have defined six boolean in order to know which component of line number description we have to print:

- `\ifl@d@pnum` for page numbers;
- `\ifl@d@ssub` for starting sub-line;
- `\ifl@d@elin` for ending line;
- `\ifl@d@esl` for ending sub-line; and
- `\ifl@d@dash` for the dash between the starting and ending groups.

There is no boolean for the line number because it is always printed.

Maieul Rouquette has added `\ifl@d@xtwoline`s and `\ifl@d@xmorethantwoline`s to print a symbol which stands for “and subsequent” when there are two, three or more lines.

```

\ifl@d@pnum
\ifl@d@ssub 2524 \newif\ifl@d@pnum
\ifl@d@elin 2525 \newif\ifl@d@ssub
\ifl@d@esl 2526 \newif\ifl@d@elin
\ifl@d@dash 2527 \newif\ifl@d@esl
\ifl@d@xtwoline 2528 \newif\ifl@d@dash
\ifl@d@xmorethantwoline 2529 \newif\ifl@d@xtwoline%
2530 \newif\ifl@d@xmorethantwoline%

\l@dparsefootspec \l@dparsefootspec{<spec>}{<lemma>}{<text>} parses a footnote specification. <lemma>
\l@dp@rsefootspec and <text> are the lemma and text respectively. <spec> is the line and page num-
\l@dparsestartpage ber and lemma font specifier in \l@d@nnums style format. The real work is done by
\l@dparsestartline \l@dp@rsefootspec which defines macros holding the numeric values. Just a reminder
\l@dparsestartsub of the arguments:
\l@dparseendpage \printlines #1 | #2 | #3 | #4 | #5 | #6 | #7
\l@dparseendline \printlines start-page | line | subline | end-page | line | subline | font
\l@dparseendsub 2531 \newcommand*{\l@dparsefootspec}[3]{\l@dp@rsefootspec#1|}
2532 \def\l@dp@rsefootspec#1|#2|#3|#4|#5|#6|#7|{%
2533 \gdef\l@dparsestartpage{#1}%
2534 \gdef\l@dparsestartline{#2}%
2535 \gdef\l@dparsestartsub{#3}%
2536 \gdef\l@dparseendpage{#4}%
2537 \gdef\l@dparseendline{#5}%
2538 \gdef\l@dparseendsub{#6}%
2539 }

Initialise the several number value macros.

2540 \def\l@dparsestartpage{0}%
2541 \def\l@dparsestartline{0}%
2542 \def\l@dparsestartsub{0}%
2543 \def\l@dparseendpage{0}%
2544 \def\l@dparseendline{0}%
2545 \def\l@dparseendsub{0}%
2546

```

`\setprintlines` The macro `\setprintlines` does the work of deciding what numbers should be printed. Its arguments are the same as the first 6 of `\printlines`.

```
2547 \newcommand*{\setprintlines}[6]{%
2548   \l@dpnumfalse \l@ddashfalse
```

We print the page numbers only if: 1) we are doing the lineation by page, and 2) the ending page number is different from the starting page number.a

```
2549   \ifbypage@
2550     \ifnum#4=#1 \else
2551       \l@dpnumtrue
2552       \l@ddashtrue
2553     \fi
2554   \fi
```

We print the ending line number if: (1) we are printing the ending page number, or (2) it is different from the starting line number.

```
2555   \ifl@dpnum \l@d@elintrue \else \l@d@elinfalse \fi
2556   \ifnum#2=#5 \else
2557     \l@d@elintrue
2558     \l@ddashtrue
2559   \fi
```

We print the starting sub-line if it is nonzero.

```
2560   \l@d@ssubfalse
2561   \ifnum#3=0 \else
2562     \l@d@ssubtrue
2563   \fi
```

We print the ending sub-line if it is nonzero and: (1) it is different from the starting sub-line number, or (2) the ending line number is being printed.

```
2564   \l@d@eslfalse
2565   \ifnum#6=0 \else
2566     \ifnum#6=#3
2567       \ifl@d@elin \l@d@esltrue \else \l@d@eslfalse \fi
2568     \else
2569       \l@d@esltrue
2570       \l@ddashtrue
2571     \fi
2572   \fi%
```

However, if the `\Xtwolines` is set for the current series, we do not print the last line number.

```
2573   \ifl@d@dash%
2574   \ifboolexpr{togl{fulllines@} or test{\ifcsemt{Xtwolines@}\@currentseries}}}%
2575     {%
2576     {%
2577       \setistwofollowinglines{#1}{#2}{#4}{#5}%
2578       \ifboolexpr{%
2579         (%
2580           togl {Xtwolinesbutnotmore@\@currentseries}%
2581           and not%
```

```

2582             (%)
2583             bool {istwofollowinglines@}%
2584             )%
2585         )%
2586     or%
2587     (%)
2588         (not test{\ifnumequal{#1}{#4}})%
2589         and togl{Xtwolinesonlyinsamepage@\@currentseries}%
2590     )%
2591 }%
2592 {}%
2593 {%
2594     \l@d@dashfalse%
2595     \l@d@Xtwolinestrue%
2596     \l@d@elinfalse%
2597     \l@d@eslfalse%
2598     \ifcempty{Xmorethantwolines@\@currentseries}%
2599     {}%
2600     {\ifistwofollowinglines@\else%
2601         \l@d@Xmorethantwolinestrue%
2602         \fi%
2603     }%
2604 }%
2605 }%
2606 \fi%
End of \setprintlines.
2607 }%

```

`\setistwofollowinglines` The `\ifistwofollowinglines` boolean, used by the `\Xtwolines` and related setting, 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 `\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.

```

2608 \newif\ifistwofollowinglines@%
2609 \newcommand{\setistwofollowinglines}[4]{%
2610     \ifcsdef{lastlinenumberon@#1}%
2611         {\numdef{tmp}{\csuse{lastlinenumberon@#1}}}%
2612         {\numdef{tmp}{0}}%
2613     \istwofollowinglines@false%
2614     \ifnumequal{#4-#2}{1}%
2615     {\istwofollowinglines@true}%
2616     {\ifbypage@%
2617         \ifnumequal{#3-#1}{1}%
2618         {%
2619             \ifnumequal{#2}{\tmp}%
2620             {\ifnumequal{#4}{1}{\istwofollowinglines@true}{}}%
2621             {}%
2622         }%
2623         {}%
2624     \fi%
2625 }%
2626 }%

```

`\printlines` So, we have decided which part of line number sets will be printed depending of these value. Now we are ready to print them. If the lineation is by pstart, we print the pstart.

```

2627 \def\printlines#1|#2|#3|#4|#5|#6|#7|{%
2628     \begingroup%

```

If we use LuaTeX, ensure we use good text's direction.

```

2629     \ifluatex%
2630     \luatextextdir TLT%
2631     \fi%

```

Decide which part of line number components we will print.

```

2632     \setprintlines{#1}{#2}{#3}{#4}{#5}{#6}%

```

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 the start line number.

```

2633     \ifdimequal{\csuse{Xboxstartlinenum@\@currentseries}}{0pt}%
2634     {\bgroup}%
2635     {\leavevmode\hbox to \csuse{Xboxstartlinenum@\@currentseries}\bgroup\hfill}%
2636     \ifl@d@pnum #1\fullstop\fi
2637     \linenumrep{#2}
2638     \ifl@d@ssub \fullstop \sublinenumrep{#3}\fi
2639     \egroup%

```

Then print the dash + end line number, or the range symbol.

```

2640     \ifdimequal{\csuse{Xboxendlinenum@\@currentseries}}{0pt}%
2641     {\bgroup}%
2642     {\hbox to \csuse{Xboxendlinenum@\@currentseries}\bgroup}%
2643     \ifl@d@Xtwolines%

```



```

2644 \ifl@d@Xmorethantwolines%
2645 \csuse{Xmorethantwolines@\@currentseries}%
2646 \else%
2647 \csuse{Xtwolines@\@currentseries}%
2648 \fi%
2649 \else%
2650 \ifl@d@dash \endashchar\fi%
2651 \ifl@d@pnum #4\fullstop\fi%
2652 \ifl@d@elin \linenumrep{#5}\fi%
2653 \ifl@d@esl \ifl@d@elin \fullstop\fi \sublinenumrep{#6}\fi%
2654 \fi%
2655 \ifdimequal{\csuse{Xboxendlinenum@\@currentseries}}{0pt}%
2656 {}%
2657 {\hfill}%Prevent underfull hbox
2658 \egroup%
2659 \endgroup%
2660 }%

```

## XIII Familiar footnotes

### XIII.1 Adjacent footnotes

The original edmac provided users with five series of critical footnotes (`\Afootnote` `\Bfootnote` `\Cfootnote` `\Dfootnote` `\Efootnote`), and  $\TeX$  provides a single numbered footnote. The `reledmac` 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 seemed to Peter Wilson such a useful ability that it was provided automatically by `eledmac`.

Maieul Rouquette has maintained this feature in `reledmac`, despite he thought that is not directly in relationship with the aim of `reledmac`.

`\multiplefootnotemarker` These macros may have been defined by the memoir class, are provided by the `footmisc` package and perhaps by other footnote packages. That is why we use `\providecommand` and not `\newcommand`.

```

2661 \providecommand*\multiplefootnotemarker}{3sp}
2662 \providecommand*\multfootsep}{\textsuperscript{\normalfont,}}
2663

```

`\m@mmf@prepare` A pair of self-cancelling kerns. This may have been defined in the memoir class.

```

2664 \providecommand*\m@mmf@prepare}{%
2665 \kern-\multiplefootnotemarker
2666 \kern\multiplefootnotemarker\relax}

```

`\m@mmf@check` This may have been defined in the memoir class. If it recognises the last kern as `\multiplefootnotemarker` it typesets `\multfootsep`.

```

2667 \providecommand*\m@mmf@check}{%
2668 \ifdim\lastkern=\multiplefootnotemarker\relax

```

```

2669 \edef\x@sf{\the\spacefactor}%
2670 \unkern
2671 \multfootsep
2672 \spacefactor\x@sf\relax
2673 \fi}
2674

```

We have to modify \@footnotetext and \@footnotemark. However, if memoir is used the modifications have already been made.

```

2675 \ifclassloaded{memoir}{}{%

```

```

\@footnotetext Add \m@mmf@prepare at the end of \@footnotetext.

```

```

2676 \apptocmd{\@footnotetext}{\m@mmf@prepare}{}{}

```

```

\@footnotemark Modify \@footnotemark to cater for adjacent \footnotes.

```

```

2677
2678 \patchcmd{\@footnotemark}
2679 {\nobreak}
2680 {\m@mmf@check
2681 \nobreak
2682 }
2683 {}{}
2684 \patchcmd{\@footnotemark}
2685 {\@makefnmark}
2686 {\@makefnmark
2687 \m@mmf@prepare
2688 }
2689 {}{}

```

Finished the modifications for the non-memoir case.

```

2690 }

```

```

2691

```

## XIII.2 Regular footnotes for numbered texts

```

\l@doldold@footnotetext In order to enable the regular \footnotes in numbered text we have to play around
\@footnotetext with its \@footnotetext, using different forms for when in numbered or regular text.

```

```

2692 \pretocmd{\@footnotetext}{%
2693 \ifnumberedpar@
2694 \edtext{}{\l@dbfnote{#1}}}%
2695 \else
2696 }{}{}
2697 \apptocmd{\@footnotetext}{\fi}{}{}%

```

```

\l@dbfnote \l@dbfnote adds the footnote to the insert list, and \vl@dbfnote calls the original
\vl@dbfnote \@footnotetext.

```

```

2698
2699 \newcommand{\l@dbfnote}[1]{%
2700   \ifnumberedpar@
2701   \gdef\@tag{#1\relax}%
2702   \ifledRcol%
2703     \xright@appenditem{\noexpand\l@dbfnote{\expandonce\@tag}}{\@thefnmark}}%
2704     \to\inserts@listR
2705     \global\advance\insert@countR \@ne%
2706   \else%
2707     \xright@appenditem{\noexpand\l@dbfnote{\expandonce\@tag}}{\@thefnmark}}%
2708     \to\inserts@list
2709     \global\advance\insert@count \@ne%
2710   \fi
2711   \fi\ignorespaces}
2712
2713 \newcommand{\vl@dbfnote}[2]{%
2714   \def\@thefnmark{#2}%
2715   \@footnotetext{#1}%
2716   }%

```

### XIII.3 Footnote formats

Some of the code for the various formats is remarkably similar to that in section ??.

The following macros generally set things up for the ‘standard’ footnote format.

`\prebodyfootmark` Two convenience macros for use by `\...@footnotemark...` macros.

```

\postbodyfootmark 2717 \newcommand*\prebodyfootmark{%
2718   \leavevmode
2719   \ifhmode
2720     \edef\@x@sf{\the\spacefactor}%
2721     \m@mmf@check
2722     \nobreak
2723   \fi}
2724 \newcommand*\postbodyfootmark{%
2725   \m@mmf@prepare
2726   \ifhmode\spacefactor\@x@sf\fi\relax}
2727

```

### XIII.4 Footnote arrangement

#### XIII.4.1 User level macro

`\arrangementX` `\arrangementX[⟨s⟩]{⟨arrangement⟩}` command calls, for each series, a specific command which set many counters and commands in order to define specific arrangement.

```

2728 \newcommandx{\arrangementX}[2][1,usedefault]{%
2729   \def\do##1{%
2730     \csname arrangementX@#2\endcsname{##1}%
2731   }%
2732   \ifstrepty{#1}%

```

```

2733    {%
2734    \dolistloop{\@series}%
2735    }%
2736    {
2737    \docsvlist{#1}%
2738    }%
2739 }%
```

### XIII.4.2 Normal footnotes

`\normal@footnotemarkX` `\normal@footnotemarkX{<series>}` sets up the typesetting of the marker at the point where the footnote is called for.

```

2740 \newcommand*{\normal@footnotemarkX}[1]{%
2741   \prebodyfootmark
2742   \@nameuse{bodyfootmark#1}%
2743   \postbodyfootmark}
2744
```

`\normalbodyfootmarkX` The `\normalbodyfootmarkX{<series>}` *really* typesets the in-text marker. The style is the normal superscript.

```

2745 \newcommand*{\normalbodyfootmarkX}[1]{%
2746   \hbox{\textsuperscript{\normalfont\@nameuse{@thefnmark#1}}}}
```

`\normalvfootnoteX` `\normalvfootnoteX{<series>}{<text>}` does the `\insert` for the `<series>` and calls the series' `\footfmt . . .` to format the `<text>`.

```

2747 \notbool{parapparatus@}{\newcommand*}{\newcommand}{\normalvfootnoteX}[2]{%
2748   \insert\@nameuse{footins#1}\bgroup
2749   \csuse{bhooknoteX@#1}
2750   \csuse{notefontsizeX@#1}
2751   \footsplitskips
2752   \ifl@dpairing\ifl@dpaging\else%
2753     \setnoteswidthliketwocolumnsX@{#1}%
2754     \fi\fi%
2755     \setnotesXpositionliketwocolumns@{#1}%
2756     \spaceskip=\z@skip \xspaceskip=\z@skip
2757     \csuse{\csuse{footnote@dir}}\@nameuse{footfmt#1}{#1}{#2}\egroup}
2758
```

`\mpnormalvfootnoteX` The minipage version.

```

2759 \newcommand*{\mpnormalvfootnoteX}[2]{%
2760   \global\setbox\@nameuse{mpfootins#1}\vbox{%
2761     \unvbox\@nameuse{mpfootins#1}
2762     \csuse{bhooknoteX@#1}
2763     \csuse{notefontsizeX@#1}
2764     \hsize\columnwidth
2765     \@parboxrestore
2766     \color@begingroup
2767     \@nameuse{footfmt#1}{#1}{#2}\color@endgroup}}
2768
```

`\normalfootfmtX` `\normalfootfmtX{<series>}{<text>}` typesets the footnote text, prepended by the marker.

```

2769 \notbool{parapparatus@}{\newcommand*{\newcommand}{\normalfootfmtX}[2]{%
2770   \ifluatex%
2771     \luatextextdir\footnote@luatextextdir%
2772     \luatexpardir\footnote@luatexpardir%
2773     \par%
2774   \fi%
2775   \protected@edef\@currentlabel{%
2776     \@nameuse{thefnmark#1}%
2777   }%
2778   \ledsetnormalparstuffX{#1}%
2779   \hangindent=\csuse{hangindentX@#1}%
2780   {{\csuse{notenumfontX@#1}\@nameuse{footfootmark#1}}\strut%
2781     #2\strut\par}}
2782

```

`\normalfootfootmarkX` `\normalfootfootmarkX{<series>}` is called by `\normalfootfmtX` to typeset the footnote marker in the footer before the footnote text.

```

2783 \newcommand*{\normalfootfootmarkX}[1]{%
2784   \textsuperscript{\@nameuse{thefnmark#1}}
2785

```

`\normalfootstartX` `\normalfootstartX{<series>}` is the `<series>` footnote starting macro used in the output routine.

```

2786 \newcommand*{\normalfootstartX}[1]{%
2787   \ifdimequal{0pt}{\prenotesX@}{}%
2788   {%
2789     \iftoggle{prenotesX@}{%
2790       \togglefalse{prenotesX@}%
2791       \skip\csname footins#1\endcsname=%
2792       \dimexpr\csuse{prenotesX@}+\csuse{afterruleX@#1}\relax%
2793     }%
2794     {}%
2795   }%
2796   \vskip\skip\csname footins#1\endcsname%
2797   \leftskip=\z@
2798   \rightskip=\z@
2799   \ifl@dpairing\else%
2800     \hsize=\old@hsize%
2801   \fi%
2802   \setnoteswidthliketwocolumnsX@{#1}%
2803   \setnotesXpositionliketwocolumns@{#1}%
2804   \print@footnoteXrule{#1}%
2805 }%
2806

```

`\normalfootnoteruleX` The rule drawn before the footnote series group.

```

2807 \let\normalfootnoteruleX=\footnoterule
2808

```

`\normalfootgroupX` `\normalfootgroupX{⟨series⟩}` sends the contents of the `⟨series⟩` insert box to the output page without alteration.

```
2809 \newcommand*{\normalfootgroupX}[1]{%
2810   \unvbox\@nameuse{footins#1}%
2811   \hsize=\old@hsize%
2812 }%
2813
```

`\mpnormalfootgroupX` The minipage version.

```
2814 \newcommand*{\mpnormalfootgroupX}[1]{%
2815   \vskip\skip\@nameuse{mpfootins#1}
2816   \ifl@dpairing\ifparledgroup%
2817     \leavevmode\marks\parledgroup@{begin}%
2818     \marks\parledgroup@series{#1}%
2819     \marks\parledgroup@type{footnoteX}%
2820   \fi\fi\normalcolor
2821   \ifparledgroup%
2822     \ifl@dpairing%
2823     \else%
2824       \setnoteswidthliketwocolumnsX@{#1}%
2825       \setnotesXpositionliketwocolumns@{#1}%
2826       \print@footnoteXrule{#1}%
2827     \fi%
2828   \else%
2829     \setnoteswidthliketwocolumnsX@{#1}%
2830     \setnotesXpositionliketwocolumns@{#1}%
2831     \print@footnoteXrule{#1}%
2832   \fi%
2833   \unvbox\@nameuse{mpfootins#1}}
2834
```

`\normalbfnoteX`

```
2835
2836 \newcommand{\normalbfnoteX}[2]{%
2837   \ifnumberedpar@
2838     \ifledRcol%
2839       \ifluatex
2840         \footnotelang@lua[R]%
2841       \fi
2842       \@ifundefined{xpg@main@language}%if polyglossia
2843       {}%
2844       {\footnotelang@poly[R]}%
2845       \protected@xdef\thisfootnote{\csuse{thefootnote#1}}%
2846       \xright@appenditem{\noexpand\vbfnoteX{#1}{#2}}{\expandonce\thisfootnote}}%
2847       \to\inserts@listR
2848       \global\advance\insert@countR \@ne%
2849     \else%
2850       \ifluatex
2851         \footnotelang@lua%
```

```

2852     \fi
2853     \@ifundefined{xpg@main@language}%if polyglossia
2854     {}%
2855     {\footnotelang@poly}%
2856     \protected@xdef\thisfootnote{\csuse{thefootnote#1}}%
2857     \xright@appenditem{\noexpand\vbfnoteX{#1}{#2}{\expandonce\thisfootnote}}%
2858     \to\inserts@list
2859     \global\advance\insert@count \@ne%
2860     \fi
2861 \fi\ignorespaces}
2862

```

\vbfnoteX

```

2863 \newcommand{\vbfnoteX}[3]{%
2864   \@namedef{@thefnmark#1}{#3}%
2865   \@nameuse{regvfootnote#1}{#1}{#2}}
2866

```

\vnumfootnoteX

```

2867 \newcommand{\vnumfootnoteX}[2]{%
2868   \ifnumberedpar@
2869     \edtext{}{\normalbfnoteX{#1}{#2}}%
2870   \else
2871     \@nameuse{regvfootnote#1}{#1}{#2}%
2872   \fi}
2873

```

`arrangementX@normal` \arrangementX@normal{<series>} initialises the settings for the <series> footnotes. This should always be called for each series.

```

2874 \newcommand*{\arrangementX@normal}[1]{%
2875   \csgdef{series@displayX#1}{normal}
2876   \expandafter\let\csname footstart#1\endcsname=\normalfootstartX
2877   \expandafter\newcount\csname prevpage#1\endcsname
2878   \@namedef{@footnotemark#1}{\normal@footnotemarkX{#1}}
2879   \@namedef{bodyfootmark#1}{\normalbodyfootmarkX{#1}}
2880   \expandafter\let\csname regvfootnote#1\endcsname=\normalvfootnoteX
2881   \expandafter\let\csname vfootnote#1\endcsname=\vnumfootnoteX
2882   \expandafter\let\csname footfmt#1\endcsname=\normalfootfmtX
2883   \@namedef{footfootmark#1}{\normalfootfootmarkX{#1}}
2884   \expandafter\let\csname footgroup#1\endcsname=\normalfootgroupX
2885   \expandafter\let\csname footnoterule#1\endcsname=\normalfootnoteruleX
2886   \count\csname footins#1\endcsname=1000
2887   \csxdef{default@footins#1}{1000}%Use to have note only for one side
2888   \dimen\csname footins#1\endcsname=\csuse{maxhnotesX@#1}
2889   \skip\csname footins#1\endcsname=\csuse{beforenotesX@#1}%
2890   \advance\skip\csname footins#1\endcsname by\csuse{afterruleX@#1}%

```

Additions for minipages.

```

2891 \ifnoledgroup@ \else%
2892   \expandafter\let\csname mpvfootnote#1\endcsname=\mpnormalvfootnoteX

```

```

2893 \expandafter\let\csname mpfootgroup#1\endcsname=\mpnormalfootgroupX
2894 \count\csname mpfootins#1\endcsname=1000
2895 \dimen\csname mpfootins#1\endcsname=\csuse{maxhnotesX@#1}
2896 \skip\csname mpfootins#1\endcsname=\csuse{beforenotesX@#1}%
2897 \advance\skip\csname mpfootins#1\endcsname by\csuse{afterruleX@#1}%
2898 \fi
2899 }
2900

```

### XIII.4.3 Two columns footnotes

The following macros set footnotes in two columns. It is assumed that the length of each footnote is less than the column width.

`\arrangementX@twocol`

```

2901 \newcommand*{\arrangementX@twocol}[1]{%
2902   \csgdef{series@displayX#1}{twocol}
2903   \expandafter\let\csname regvfootnote#1\endcsname=\twocolvfootnoteX
2904   \expandafter\let\csname footfmt#1\endcsname=\twocolfootfmtX
2905   \expandafter\let\csname footgroup#1\endcsname=\twocolfootgroupX
2906   \dimen\csname footins#1\endcsname=\csuse{maxhnotesX@#1}%
2907   \skip\csname footins#1\endcsname=\csuse{beforenotesX@#1}%
2908   \advance\skip\csname footins#1\endcsname by \csuse{afterruleX@#1}\relax%
2909   \twocolfootsetupX{#1}
2910   \ifnoledgroup@ \else%
2911     \expandafter\let\csname mpvfootnote#1\endcsname=\mpnormalvfootnoteX
2912     \expandafter\let\csname mpfootgroup#1\endcsname=\mptwocolfootgroupX
2913     \skip\csname mpfootins#1\endcsname=\csuse{beforenotesX@#1}%
2914     \advance\skip\csname mpfootins#1\endcsname by\csuse{afterruleX@#1}
2915     \mptwocolfootsetupX{#1}
2916   \fi%
2917 }
2918

```

`\twocolfootsetupX \twocolfootsetupX{<series>}`

```

\mptwocolfootsetupX 2919 \newcommand*{\twocolfootsetupX}[1]{%
2920   \count\csname footins#1\endcsname 500
2921   \csxdef{default@footins#1}{500}%Use this to confine the notes to one side only
2922   \multiply\dimen\csname footins#1\endcsname by \tw@
2923   \newcommand*{\mptwocolfootsetupX}[1]{%
2924     \count\csname mpfootins#1\endcsname 500
2925     \multiply\dimen\csname mpfootins#1\endcsname by \tw@
2926

```

`\twocolvfootnoteX \twocolvfootnoteX{<series>}`

```

2927 \notbool{parapparatus@}{\newcommand*}{\newcommand*}{\twocolvfootnoteX}[2]{%
2928   \insert\csname footins#1\endcsname\bgroup
2929     \csuse{notefontsizeX@#1}
2930     \footssplit skips

```



```

2931 \spaceskip=\z@skip \xspaceskip=\z@skip
2932 \@nameuse{footfmt#1}{#1}{#2}\egroup
2933

```

```

\twocolfootfmtX \twocolfootfmtX{<series>}
2934 \notbool{parapparatus@}{\newcommand*}{\newcommand}{\twocolfootfmtX}[2]{%
2935 \protected@edef\@currentlabel{%
2936 \@nameuse{thefnmark#1}%
2937 }%
2938 \normal@pars
2939 \hangindent=\csuse{hangindentX@#1}%
2940 \hsize \csuse{hsizetwocolX@#1}
2941 \nottoggle{parindentX@#1}{\parindent=\z@}{%
2942 \tolerance=5000\relax
2943 \leavevmode
2944 \csuse{colalignX@#1}%
2945 {\csuse{notenumberX@#1}\@nameuse{footfootmark#1}\strut%
2946 #2\strut\par}\allowbreak}
2947

```

```

\twocolfootgroupX \twocolfootgroupX{<series>}
\mptwocolfootgroupX 2948 \newcommand*{\twocolfootgroupX}[1]{\csuse{notefontsizeX@#1}
2949 \splittopskip=\ht\strutbox
2950 \expandafter
2951 \rigidbalance\csname footins#1\endcsname \tw@ \splittopskip}}
2952 \newcommand*{\mptwocolfootgroupX}[1]{%
2953 \vskip\skip\@nameuse{mpfootins#1}
2954 \ifl@dpairing\ifparledgroup%
2955 \leavevmode\marks\parledgroup@{begin}%
2956 \marks\parledgroup@series{#1}%
2957 \marks\parledgroup@type{footnoteX}%
2958 \fi\fi\normalcolor
2959 \ifparledgroup%
2960 \ifl@dpairing%
2961 \else%
2962 \setnoteswidthliketwocolumnsX@{#1}%
2963 \setnotesXpositionliketwocolumns@{#1}%
2964 \print@footnoteXrule{#1}%
2965 \fi%
2966 \else%
2967 \setnoteswidthliketwocolumnsX@{#1}%
2968 \setnotesXpositionliketwocolumns@{#1}%
2969 \print@footnoteXrule{#1}%
2970 \fi%
2971 \splittopskip=\ht\strutbox
2972 \expandafter
2973 \rigidbalance\csname mpfootins#1\endcsname \tw@ \splittopskip}}
2974

```

### XIII.4.4 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.

```
\arrangementX@threecol
2975 \newcommand*{\arrangementX@threecol}[1]{%
2976   \csgdef{series@displayX#1}{threecol}
2977   \expandafter\let\csname regvfootnote#1\endcsname=\threecolvfootnoteX
2978   \expandafter\let\csname footfmt#1\endcsname=\threecolfootfmtX
2979   \expandafter\let\csname footgroup#1\endcsname=\threecolfootgroupX
2980   \dimen\csname footins#1\endcsname=\csuse{maxhnotesX@#1}%
2981   \skip\csname footins#1\endcsname=\csuse{beforenotesX@#1}%
2982   \advance\skip\csname footins#1\endcsname by \csuse{afterruleX@#1}\relax%
2983   \threecolfootsetupX{#1}
2984   \ifnoledgroup\else%
2985     \expandafter\let\csname mpvfootnote#1\endcsname=\mpnormalvfootnoteX
2986     \expandafter\let\csname mpfootgroup#1\endcsname=\mpthreecolfootgroupX
2987     \skip\csname mpfootins#1\endcsname=\csuse{beforenotesX@#1}%
2988     \advance\skip\csname mpfootins#1\endcsname by \csuse{afterruleX@#1}
2989     \mpthreecolfootsetupX{#1}
2990   \fi%
2991 }
2992

\threecolfootsetupX \threecolfootsetupX{<series>}
\mpthreecolfootsetupX
2993 \newcommand*{\threecolfootsetupX}[1]{%
2994   \count\csname footins#1\endcsname 333
2995   \csxdef{default@footins#1}{333}%Use this to confine the notes to one side only
2996   \multiply\dimen\csname footins#1\endcsname by \thr@@
2997 \newcommand*{\mpthreecolfootsetupX}[1]{%
2998   \count\csname mpfootins#1\endcsname 333
2999   \multiply\dimen\csname mpfootins#1\endcsname by \thr@@
3000

\threecolvfootnoteX \threecolvfootnoteX{<series>}{<text>}
3001 \notbool{parapparatus@}{\newcommand*}{\newcommand}{\threecolvfootnoteX}[2]{%
3002   \insert\csname footins#1\endcsname\bgroup
3003     \csuse{notefontsizeX@#1}
3004     \footsplitskips
3005     \@nameuse{footfmt#1}{#1}{#2}\egroup}
3006

\threecolfootfmtX \threecolfootfmtX{<series>}
3007 \notbool{parapparatus@}{\newcommand*}{\newcommand}{\threecolfootfmtX}[2]{%
3008   \protected@edef\@currentlabel{%
3009     \@nameuse{@thefnmark#1}%
3010   }%
3011   \hangindent=\csuse{hangindentX@#1}%

```

```

3012 \normal@pars
3013 \hsize \csuse{hsizethreecolX@#1}
3014 \nottoggle{parindentX@#1}{\parindent=\z@}{\ %
3015 \tolerance=5000\relax
3016 \leavevmode
3017 \csuse{colalignX@#1}%
3018 {\csuse{notenumfontX@#1}\@nameuse{footfootmark#1}\strut%
3019 #2\strut\par}\allowbreak}
3020

```

```

\threecolfootgroupX \threecolfootgroupX{<series>}
\mpthreecolfootgroupX
3021 \newcommand*{\threecolfootgroupX}[1]{\csuse{notefontsizeX@#1}
3022 \splittopskip=\ht\strutbox
3023 \expandafter
3024 \rigidbalance\csname footins#1\endcsname \thr@@ \splittopskip}}
3025 \newcommand*{\mpthreecolfootgroupX}[1]{\%
3026 \vskip\skip\@nameuse{mpfootins#1}
3027 \ifl@dpairing\ifparledgroup
3028 \leavevmode\marks\parledgroup@{begin}%
3029 \marks\parledgroup@series{#1}%
3030 \marks\parledgroup@type{footnoteX}%
3031 \fi\fi\normalcolor
3032 \ifparledgroup%
3033 \ifl@dpairing%
3034 \else%
3035 \setnoteswidthliketwocolumnsX@{#1}%
3036 \setnotesXpositionliketwocolumns@{#1}%
3037 \print@footnoteXrule{#1}%
3038 \fi%
3039 \else%
3040 \setnoteswidthliketwocolumnsX@{#1}%
3041 \setnotesXpositionliketwocolumns@{#1}%
3042 \print@footnoteXrule{#1}%
3043 \fi%
3044 \splittopskip=\ht\strutbox
3045 \expandafter
3046 \rigidbalance\csname mpfootins#1\endcsname \thr@@ \splittopskip}}
3047

```

#### XIII.4.5 Paragraphed footnotes

The following macros set footnotes as one paragraph.

```

\arrangementX@threecol \footparagraphX{<series>}
3048 \newcommand*{\arrangementX@paragraph}[1]{\%
3049 \csgdef{series@displayX#1}{paragraph}%
3050 \expandafter\newcount\csname #1prevpage@num\endcsname
3051 \expandafter\let\csname footstart#1\endcsname=\parafootstartX
3052 \expandafter\let\csname regvfootnote#1\endcsname=\para@vfootnoteX

```

```

3053 \expandafter\let\csname footfmt#1\endcsname=\parafootfmtX
3054 \expandafter\let\csname footgroup#1\endcsname=\para@footgroupX
3055 \expandafter\let\csname footnoterule#1\endcsname=\normalfootnoteruleX
3056 \count\csname footins#1\endcsname=1000
3057 \csxdef{default@footins#1}{1000}%Use this to confine the notes to one side only
3058 \dimen\csname footins#1\endcsname=\csuse{maxhnotesX@#1}
3059 \skip\csname footins#1\endcsname=\csuse{beforenotesX@#1}%
3060 \advance\skip\csname footins#1\endcsname by\csuse{afterruleX@#1}%
3061 \para@footsetupX{#1}
3062 \ifnoledgroup\else
3063   \expandafter\let\csname mpvfootnote#1\endcsname=\mppara@vfootnoteX
3064   \expandafter\let\csname mpfootgroup#1\endcsname=\mppara@footgroupX
3065   \count\csname mpfootins#1\endcsname=1000
3066   \dimen\csname mpfootins#1\endcsname=\csuse{maxhnotesX@#1}
3067   \skip\csname mpfootins#1\endcsname=\csuse{beforenotesX@#1}%
3068   \advance\skip\csname mpfootins#1\endcsname by\csuse{afterruleX@#1}%
3069 \fi
3070 }
3071
\para@footsetupX \para@footsetupX{<series>}
3072 \newcommand*{\para@footsetupX}[1]{\csuse{notefontsizeX@#1}
3073   \setnoteswidthliketwocolumnsX@{#1}%
3074   \dimen0=\baselineskip
3075   \multiply\dimen0 by 1024
3076   \divide\dimen0 by \columnwidth \multiply\dimen0 by \footfudgefiddle\relax%
3077   \expandafter
3078   \xdef\csname footfudgefactor#1\endcsname{%
3079     \expandafter\strip@pt\dimen0 }}
3080
\parafootstartX \parafootstartX{<series>}
3081 \newcommand*{\parafootstartX}[1]{%
3082   \ifdimequal{0pt}{\prenotesX@}{}%
3083   {%
3084     \iftoggle{prenotesX@}{%
3085       \togglefalse{prenotesX@}%
3086       \skip\csname footins#1\endcsname=%
3087         \dimexpr\csuse{prenotesX@}+\csuse{afterruleX@#1}\relax%
3088     }%
3089   }%
3090   }%
3091   \vskip\skip\csname footins#1\endcsname%
3092   \leftskip=\z@
3093   \rightskip=\z@
3094   \parindent=\z@
3095   \vskip\skip\@nameuse{footins#1}%
3096   \setnoteswidthliketwocolumnsX@{#1}%
3097   \setnotesXpositionliketwocolumns@{#1}%
3098   \print@footnoteXrule{#1}%

```

3099 }  
3100

```
\para@vfootnoteX \para@vfootnoteX{<series>}{<text>}
\mppara@vfootnoteX 3101 \newcommand*{\para@vfootnoteX}[2]{%
3102 \insert\csname footins#1\endcsname
3103 \bgroup
3104 \csuse{bhooknoteX@#1}
3105 \csuse{notefontsizeX@#1}
3106 \footsplitskips
3107 \setbox0=\vbox{\hsize=\maxdimen
3108 \noindent\@nameuse{footfmt#1}{#1}{#2}}%
3109 \setbox0=\hbox{\unvxhX{0}{#1}}%
3110 \dp0=\z@
3111 \ht0=\csname footfudgefactor#1\endcsname\wd0
3112 \box0
3113 \penalty0
3114 \egroup}
3115 \newcommand*{\mppara@vfootnoteX}[2]{%
3116 \global\setbox\@nameuse{mpfootins#1}\vbox{%
3117 \unvbox\@nameuse{mpfootins#1}
3118 \csuse{bhooknoteX@#1}
3119 \csuse{notefontsizeX@#1}
3120 \footsplitskips
3121 \setbox0=\vbox{\hsize=\maxdimen
3122 \noindent\color@begingroup\@nameuse{footfmt#1}{#1}{#2}\color@endgroup}%
3123 \setbox0=\hbox{\unvxhX{0}{#1}}%
3124 \dp0=\z@
3125 \ht0=\csname footfudgefactor#1\endcsname\wd0
3126 \box0
3127 \penalty0}}
3128
```

\unvxhX

```
3129 \newcommand*{\unvxhX}[2]{% 2th is optional for retro-compatibility
3130 \setbox0=\vbox{\unvbox#1%
3131 \global\setbox1=\lastbox}%
3132 \unhbox1
3133 \unskip % remove \rightskip,
3134 \unskip % remove \parfillskip,
3135 \unpenalty % remove \penalty of 10000,
3136 \hskip\csuse{afternoteX@#2}} % but add the glue to go between the notes
3137
```

```
\parafootfmtX \parafootfmtX{<series>}
3138 \newcommand*{\parafootfmtX}[2]{%
3139 \protected@edef\@currentlabel{%
3140 \@nameuse{@thefnmark#1}}%
3141 }%
```

```

3142 \insertparafootsepX{#1}%
3143 \ledsetnormalparstuffX{#1}%
3144 {\csuse{notenumfontX@#1}%
3145 \csuse{notenumfontX@#1}%
3146 \@nameuse{footfootmark#1}%
3147 \strut%
3148 #2\penalty-10}}
3149

\para@footgroupX \para@footgroupX{<series>}
\mppara@footgroupX 3150 \newcommand*{\para@footgroupX}[1]{%
3151 \unvbox\csname footins#1\endcsname
3152 \ifcsstring{raggedX@#1}{L}{\RaggedLeft}{}%
3153 \ifcsstring{raggedX@#1}{R}{\RaggedRight}{}%
3154 \makehboxofhboxes
3155 \setbox0=\hbox{\unhbox0 \removehboxes}%
3156 \csuse{notefontsizeX@#1}
3157 \noindent\unhbox0\par}
3158 \newcommand*{\mppara@footgroupX}[1]{%
3159 \setnoteswidthliketwocolumnsX@{#1}%
3160 \vskip\skip\@nameuse{mpfootins#1}
3161 \ifl@dpairing\ifparledgroup
3162 \leavevmode%
3163 \leavevmode\marks\parledgroup@{begin}%
3164 \marks\parledgroup@series{#1}%
3165 \marks\parledgroup@type{footnoteX}%
3166 \fi\fi\normalcolor
3167 \ifparledgroup%
3168 \ifl@dpairing%
3169 \else%
3170 \setnoteswidthliketwocolumnsX@{#1}%
3171 \setnotesXpositionliketwocolumns@{#1}%
3172 \print@footnoteXrule{#1}%
3173 \fi%
3174 \else%
3175 \setnoteswidthliketwocolumnsX@{#1}%
3176 \setnotesXpositionliketwocolumns@{#1}%
3177 \print@footnoteXrule{#1}%
3178 \fi%
3179 \unvbox\csname mpfootins#1\endcsname
3180 \ifcsstring{raggedX@#1}{L}{\RaggedLeft}{}%
3181 \ifcsstring{raggedX@#1}{R}{\RaggedRight}{}%
3182 \makehboxofhboxes
3183 \setbox0=\hbox{\unhbox0 \removehboxes}%
3184 \csuse{notefontsizeX@#1}
3185 \noindent\unhbox0\par}}
3186

```

**Insertion of the footnotes separator** The command `\insertparafootsepX{<series>}` must be called at the beginning of `\parafootftmX`.

```
\prevpage@num
\Xinsertparafootsep 3187 \newcommand{\insertparafootsepX}[1]{%
3188   \ifnumequal{\csuse{prevpage#1@num}}{\page@num}%
3189   {\csuse{parafootsepX@#1}}%
3190   }%
3191 }
```

## XIV Code common to both critical and familiar footnote in normal arrangement

`\par` should always be redefined to `\endgraf` within the format macro (this is what `\normal@pars` does), to override tricky material in the main text to get the lines numbered automatically (as set up by `\autopar`, for example).

In the case of footnote arranged in a “normal” way, we also must set some setting for paragraph indent and text direction when using `LuaLaTeX`.

That why we have defined `\ledsetnormalparstuff@common` in order to make this setting for both familiar and critical notes. This command is called by command to make specific setting to critical or familiar footnote.

```
dsetnormalparstuff@common
\Xledsetnormalparstuff 3192 \newcommand*{\ledsetnormalparstuff@common}{%
\ledsetnormalparstuffX 3193   \ifluatex%
3194     \luatextextdir\footnote@luatextextdir%
3195     \luatexpardir\footnote@luatexpardir%
3196     \fi%
3197     \csuse{\csuse{footnote@dir}}%
3198     \normal@pars%
3199     \parfillskip \z@ \@plus 1fil}%
3200
3201 \newcommand*{\Xledsetnormalparstuff}[1]{%
3202   \ledsetnormalparstuff@common%
3203   \nottoggle{Xparindent@#1}{\noindent}}{\noindent and and not \parindent=0pt to avoid to break the (bad)
3204 }%
3205
3206 \newcommand*{\ledsetnormalparstuffX}[1]{%
3207   \ledsetnormalparstuff@common%
3208   \nottoggle{parindentX@#1}{\noindent}}{\noindent and and not \parindent=0pt to avoid to break the (bad)
3209 }%
```

## XV Footnotes’ width for two columns

We define here some commands which make sense only with `reledpar`, 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.

`\old@hsize` These two commands are called at the beginning of critical or familiar notes groups.  
`\setXnoteswidthliketwocolumns@` They set, if the option is enabled, the `\hsize`. They are also called at the on the setup  
`\setnoteswidthliketwocolumnsX@` for paragraphed notes.

```

3210
3211 \newdimen\old@hsize%
3212 \AtBeginDocument{\old@hsize=\hsize}%
3213
3214 \newcommand{\setXnoteswidthliketwocolumns@}[1]{%
3215   \global\let\hsize@fornote=\hsize%
3216   \global\old@hsize=\hsize%
3217   \iftoggle{Xnoteswidthliketwocolumns@#1}{%
3218     {%
3219       \csuse{setwidthliketwocolumns@\columns@position}%
3220       \global\let\hsize@fornote=\hsize%
3221     }%
3222   }%
3223   \let\hsize=\hsize@fornote%
3224   \let\columnwidth=\hsize@fornote%
3225 }%
3226
3227 \newcommand{\setnoteswidthliketwocolumnsX@}[1]{%
3228   \global\let\hsize@fornote=\hsize%
3229   \global\old@hsize=\hsize%
3230   \iftoggle{noteswidthliketwocolumnsX@#1}{%
3231     {%
3232       \csuse{setwidthliketwocolumns@\columns@position}%
3233       \global\let\hsize@fornote=\hsize%
3234     }%
3235   }%
3236   \let\hsize=\hsize@fornote%
3237   \let\columnwidth=\hsize@fornote%
3238 }%
3239

```

`\setXnotespositionliketwocolumns@` These two commands set the position of the critical / familiar footnotes, depending on  
`\setnotesXpositionliketwocolumns@` the hooks `Xnoteswidthliketwocolumns` and `noteswidthliketwocolumnsX`. They  
 call commands which are defined only in `reledpar`, because this feature has no sens  
 without `reledpar`.

```

3240 \newcommand{\setXnotespositionliketwocolumns@}[1]{%
3241   \iftoggle{Xnoteswidthliketwocolumns@#1}{%
3242     \csuse{setnotespositionliketwocolumns@\columns@position}%
3243   }{}%
3244 }%
3245
3246 \newcommand{\setnotesXpositionliketwocolumns@}[1]{%
3247   \iftoggle{noteswidthliketwocolumnsX@#1}{%
3248     \csuse{setnotespositionliketwocolumns@\columns@position}%
3249   }{}%
3250 }%

```



3251

## XVI Footnotes' order

`\fnpos` The `\fnpos` and `\mpfnpos` simply place their arguments in `\@fnpos` and `\@mpfnpos`, which will be used later in the output routine.

```

\@fnpos 3252 \def\@fnpos{familiar-critical}
\@mpfnpos 3253 \def\@mpfnpos{critical-familiar}
3254 \newcommand{\fnpos}[1]{\xdef\@fnpos{#1}}
3255 \newcommand{\mpfnpos}[1]{\xdef\@mpfnpos{#1}}
```

## XVII Footnotes' rule

Because the footnotes' rules can be shifted to the right when footnotes are set like two columns, we do not print them directly, but we put them in a `\vbox`.

```

\print@Xfootnoterule
\print@footnoteXrule 3256 \newcommand{\print@Xfootnoterule}[1]{%
3257 \vskip-\csuse{Xafterrule@#1}%Because count in \dimen\csuse{#1footins}
3258 \nointerlineskip%
3259 \moveleft-\leftskip\vbox{\csuse{#1footnoterule}}%
3260 \nointerlineskip%
3261 \vskip\csuse{Xafterrule@#1}%
3262 }%
3263
3264 \newcommand{\print@footnoteXrule}[1]{%
3265 \vskip-\csuse{afterruleX@#1}%Because count in \dimen\csuse{footins#1}
3266 \nointerlineskip%
3267 \moveleft-\leftskip\vbox{\csuse{footnoterule#1}}%
3268 \nointerlineskip%
3269 \vskip\csuse{afterruleX@#1}%
3270 }%
3271
```

## XVIII Specific skip for first series of footnotes

### XVIII.0.1 Overview

`\Xbeforenotes` inserts a specific skip for the first series of notes in a page. As we can't 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 `\Xbeforenotes`. It also 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 skip of the current series to the value normally used by the series which was marked as the first of the page. It also 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 `\Xafterrule` 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.

And now, implementation !

### XVIII.0.2 User level command

`\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.

```
3272 \newtoggle{preXnotes@}
3273 \toggletrue{preXnotes@}
3274 \newcommand{\preXnotes@}{0pt}
3275 \newcommand*{\preXnotes}[1]{\renewcommand{\preXnotes@}{#1}}
```

The same, but for familiar footnotes.

```
\preXnotes
\preXnotes@ 3276 \newtoggle{prenotesX@}
3277 \toggletrue{prenotesX@}
3278 \newcommand{\prenotesX@}{0pt}
3279 \newcommand*{\prenotesX}[1]{\renewcommand{\prenotesX@}{#1}}
```

### XVIII.0.3 Internal commands

```
firstXseries@
prepare@preXnotes 3280 \gdef\firstXseries@{}
3281 \newcommand{\prepare@preXnotes}[1]{%
3282   \ifdimequal{0pt}{\preXnotes@}%
3283   {%
3284     {%
3285       \IfStrEq{\firstXseries@}{}{%
3286         \global\skip\csuse{#1footins}=\preXnotes@%
3287         \global\advance\skip\csname #1footins\endcsname by\csuse{Xafterrule@#1}%

```

```

3288     \gdef\firstXseries@{#1}%
3289   }%
3290   {%
3291     \ifseriesbefore{#1}{\firstXseries@}%
3292     {%
3293       \global\skip\csuse{#1footins}=\csuse{Xbeforenotes@\firstXseries@}%
3294       \global\advance\skip\csname #1footins\endcsname by\csuse{Xafterrule@#1}%
3295       \gdef\firstXseries@{#1}%
3296     }%
3297   {}%
3298 }%
3299 }%
3300 }

```

The same thing is required for familiar notes and `\prenotesX`.

```

firstseriesX@
prepare@prenotesX 3301 \gdef\firstseriesX@{
3302 \newcommand{\prepare@prenotesX}[1]{%
3303   \ifdimequal{0pt}{\prenotesX@}%
3304   {}%
3305   {%
3306     \IfStrEq{\firstseriesX@}{}%
3307     \global\skip\csuse{footins#1}=\prenotesX@%
3308     \global\advance\skip\csname footins#1\endcsname by\csuse{afterruleX@#1}%
3309     \gdef\firstseriesX@{#1}%
3310   }%
3311   {%
3312     \ifseriesbefore{#1}{\firstseriesX@}%
3313     {%
3314       \global\skip\csuse{footins#1}=\csuse{beforenotesX@\firstseriesX@}%
3315       \global\advance\skip\csname footins#1\endcsname by\csuse{afterruleX@#1}%
3316       \gdef\firstXseries@{#1}%
3317     }%
3318   {}%
3319 }%
3320 }%
3321 }

```

## XIX Endnotes

First, check the `noend` option.

```

3322 \ifbool{noend@}{}%Used instead of \ifnoend@ to prevent expansion problem

```

```

\l@dend@open \l@dend@open and \l@dend@close are the macros that are used to open and close the
\l@dend@close endnote file. Note that all our writing to this file is \immediate: all page and line num-
bers for the endnotes are generated by the same mechanism we use for the footnotes, so
that there is no need to defer any writing to catch information from the output routine.
The argument of these two command is the series letter.

```

```

3323 \newcommand{\l@dend@open}[1]{%
3324   \global\booltrue{l@dend@#1}%
3325   \expandafter\immediate%
3326   \expandafter\openout%
3327   \csname l@d@#1end\endcsname%
3328   =\jobname.#1end\relax%
3329 }%
3330 \newcommand{\l@dend@close}[1]{%
3331   \global\boolfalse{l@dend@#1}%
3332   \expandafter\immediate%
3333   \expandafter\closeout\csname l@d@#1end\endcsname%
3334 }%
3335

```

`\l@dend@stuff` `\l@dend@stuff` is used by `\beginnumbering` to do everything that is 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.

```

3336 \newcommand{\l@dend@stuff}{%
3337   \def\do##1{%
3338     \ifbool{l@dend@##1}{}%
3339     {\l@dend@open{##1}}%
3340     \expandafter\immediate\expandafter\write\csname l@d@##1end\endcsname{\string\l@d@section}
3341   }%
3342   \dolistloop{@series}%
3343 }%
3344

```

`\endprint` The `\endprint` here is nearly identical in its functioning to `\normalfootfmt`.  
`\l@d@section` The endnote file also contains `\l@d@section` commands, which supply the section numbers from the main text; standard `reledmac` does nothing with this information, but it is 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 `\Xendnote`.

```

3345 \global\notbool{parapparatus@}{\long}\def\endprint#1#2#3#4#5{
3346   \ifXendinsertsep%
3347     \hskip\csuse{Xendafternote@#4}%
3348     \csuse{Xendsep@#4}%
3349   \else%
3350     \iftoggle{Xendparagraph@#4}%
3351       {\global\Xendinsertsep@true}%
3352     }%

```

```

3353 \fi%
3354 \xdef\currentseries{#4}%
3355 \def\do##1{%
3356   \toggletrue{##1}%
3357 }%
3358 \notblank{#5}{\docsvlist{#5}}{}%
3359 \csuse{Xendbhooknote@#4}%
3360 \csuse{Xendnotefontsize@#4}%
3361 \ifboolexpr{%
3362   togl {nonum@}%
3363   or togl {Xendnonumber@#4}%
3364 }%
3365 {\hspace{\csuse{Xendinplaceofnumber@#4}}}%
3366 {\printlineendnotearea{#1}{#4}}%
3367 \nottoggle{Xendlemmadisablefontselection@#4}%
3368   {\select@lemmafont#1|#2}%
3369   {#2}%
3370 \ifboolexpr{%
3371   togl {nosep@}%
3372   or test{\ifcsemtty{Xendlemmaseparator@#4}}%
3373 }%
3374 {\hskip\csuse{Xendinplaceoflemmaseparator@#4}}%
3375 {\nobreak%
3376   \hskip\csuse{Xendbeforelemmaseparator@#4}%
3377   \csuse{Xendlemmaseparator@#4}%
3378   \hskip\csuse{Xendafterlemmaseparator@#4}%
3379 }%
3380 #3%
3381 \nottoggle{Xendparagraph@#4}{\par}{}%
3382 \def\do##1{%
3383   \togglefalse{##1}%
3384 }%
3385 \notblank{#5}{\docsvlist{#5}}{}%
3386 }}%
3387
3388 \let\l@d@section=\@gobble
3389

```

`\printlineendnotearea` 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 `\endprint` depending of the options about repeating line numbers. The first argument is line information, the second is the notes series (A, B, C, etc.)

```

3390 \newcommand{\printlineendnotearea}[2]{%
3391   \bgroup%
3392   \csuse{Xendnotenumfont@#2}%
3393   \ifdimequal{\csuse{Xendboxlinenum@#2}}{0pt}%
3394     {\printendlines#1}%
3395     {\leavevmode%
3396       \hbox to \csuse{Xendboxlinenum@#2}%
3397       {%

```

```

3398      \IfSubStr{RC}{\csuse{Xendboxlinenumalign@#2}}{\hfill}{}%
3399      \printendlines#1|%
3400      \IfSubStr{LC}{\csuse{Xendboxlinenumalign@#2}}{\hfill}{}%
3401      }}%
3402  \egroup%
3403  \enspace%
3404 }%

```

`\setprintendlines` The `\printendlines` 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.

```

3405 \newcommand*{\setprintendlines}[6]{%
3406   \l@dpnumfalse \l@ddashfalse
3407   \ifnum#4=#1 \else
3408     \l@dpnumtrue
3409     \l@ddashtrue
3410   \fi

```

We print the ending line number if: (1) we are printing the ending page number, or (2) it is different from the starting line number.

```

3411   \ifl@dpnum \l@d@elintrue \else \l@d@elinfalse \fi
3412   \ifnum#2=#5 \else
3413     \l@d@elintrue
3414     \l@d@dashtrue
3415   \fi

```

We print the starting sub-line if it is nonzero.

```

3416   \l@d@ssubfalse
3417   \ifnum#3=0 \else
3418     \l@d@ssubtrue
3419   \fi

```

We print the ending sub-line if it is nonzero and: (1) it is different from the starting sub-line number, or (2) the ending line number is being printed.

```

3420   \l@d@eslfalse
3421   \ifnum#6=0 \else
3422     \ifnum#6=#3
3423       \ifl@d@elin \l@d@esltrue \else \l@d@eslfalse \fi
3424     \else
3425       \l@d@esltrue
3426       \l@d@dashtrue
3427     \fi
3428   \fi%

```

```

3429 \ifl@d@dash%
3430 \ifboolexpr{togl{fulllines@} or test{\ifcsempy{Xendtwolines@\@currentseries}}}%
3431 {}%
3432 {%
3433 \setistwofollowinglines{#1}{#2}{#4}{#5}%
3434 \ifboolexpr{%
3435 (%
3436     togl {Xendtwolinesbutnotmore@\@currentseries}%
3437     and not%
3438     (%
3439         bool {istwofollowinglines@}%
3440     )%
3441 )%
3442 or%
3443 (%
3444     (not test{\ifnumequal{#1}{#4}})%
3445     and togl{Xendtwolinesonlyinsamepage@\@currentseries}%
3446 )%
3447 }%
3448 {}%
3449 {%
3450 \l@d@d@dashfalse%
3451 \l@d@Xtwolinestrue%
3452 \l@d@elinfalse%
3453 \l@d@eslfalse%
3454 \ifcsempy{Xendmorethantwolines@\@currentseries}%
3455 {}%
3456 {\ifistwofollowinglines@\else%
3457     \l@d@Xmorethantwolinestrue%
3458     \fi%
3459 }%
3460 }%
3461 }%
3462 \fi%

End of \setprintendlines.
3463 }%

```

`\printendlines` Now we are ready to print it all.

```

3464 \def\printendlines#1|#2|#3|#4|#5|#6|#7|{\begingroup
3465 \setprintendlines{#1}{#2}{#3}{#4}{#5}{#6}%

```

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.

```

3466 \ifdimequal{\csuse{Xendboxstartlinenum@\@currentseries}}{0pt}%
3467 {\bgroup}%
3468 {\leavevmode\hbox to \csuse{Xendboxstartlinenum@\@currentseries}\bgroup\hfill}%

```

```

3469 \printnpnum{#1}%
3470 \linenumrep{#2}%
3471 \ifl@d@ssub \fullstop \sublinenumrep{#3}\fi
3472 \egroup%

```

And now, print the dash + the end line number, or the line number range symbol.

```

3473 \ifdimequal{\csuse{Xendboxendlinenum@\@currentseries}}{0pt}%
3474 {\bgroup}%
3475 {\hbox to \csuse{Xendboxendlinenum@\@currentseries}\bgroup}%
3476 \ifl@d@Xtwolines%
3477 \ifl@d@Xmorethantwolines%
3478 \csuse{Xendmorethantwolines@\@currentseries}%
3479 \else%
3480 \csuse{Xendtwolines@\@currentseries}%
3481 \fi%
3482 \else%
3483 \ifl@d@dash \endashchar\fi%
3484 \ifl@d@pnum \printnpnum{#4}\fi%
3485 \ifl@d@elin \linenumrep{#5}\fi%
3486 \ifl@d@esl \ifl@d@elin \fullstop\fi \sublinenumrep{#6}\fi%
3487 \fi%
3488 \ifdimequal{\csuse{Xendboxendlinenum@\@currentseries}}{0pt}%
3489 {}%
3490 {\hfill}%Prevent underfull hbox
3491 \egroup%
3492 \endgroup%
3493 }%
3494

```

`\printnpnum` A macro to print a page number in an endnote.

```

3495 \newcommand*{\printnpnum}[1]{p.#1) }
3496

```

`\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.

```

3497 \newif\ifXendinsertsep@%
3498 \newcommand*{\doendnotes}[1]{%
3499 \l@dend@close{#1}%
3500 \beginingroup
3501 \makeatletter
3502 \expandafter\let\csname #1end\endcsname=\endprint
3503 \input\jobname.#1end%
3504 \global\Xendinsertsep@false%
3505 \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.

```

3506 \newcommand*{\doendnotesbysection}[1]{%
3507 \l@nd@close{#1}%
3508 \global\expandafter\advance\csname #1end@bysection\endcsname by 1%
3509 \beginingroup%
3510 \makeatletter%
3511 \def\l@d@section##1{%
3512 \ifnumequal{##1}{\csname #1end@bysection\endcsname}%
3513 {\cslet{#1end}{\endprint}}%
3514 {\cslet{#1end}{\@gobblefive}}%
3515 }%
3516 \input\jobname.#1end%
3517 \global\Xendinsertsep@false%
3518 \endgroup%
3519 }%

End of section for end notes
3520 }%
```

## XX Generate series of notes

In this section, X means the name of the series (A, B etc.)

`\series` `\series` creates one more new series. It is a public command, which just loops on the private command `\newseries@`.

```

3521 \newcommand{\newseries}[1]{%
3522 \def\do##1{\newseries@{##1}}%
3523 \docsvlist{#1}
3524 }
```

`\@series` The `\series@` macro is an etoolbox list, which contains the name of all series.

```

3525 \newcommand{\@series}{}
```

The command `\newseries@` creates a new series of the footnote.

`\newseries@`

```

3526 \newcommand{\newseries@}[1]{
```

### XX.1 Test if series is still existing

```

3527 \xifinlist{#1}{\@series}{\led@warn@SeriesStillExist{#1}}%
3528 {%
```

### XX.2 Init specific to reledpar

When calling `\newseries@` after having loaded `reledpar`, we need to load specific setting.

```

3529 \ifdefined\newseries@par%
3530 \newseries@par{#1}%
3531 \fi%

```

### XX.3 For critical footnotes

Critical footnotes are those which start with letters. We look for the `\nocritical` option of `reledmac`.

```

3532 \unless\ifnocritical@

```

#### XX.3.1 Options

```

3533 \newtoggle{Xparindent@#1}
3534 \newtoggle{Xlemmadisablefontselection@#1}
3535 \csgdef{Xhangindent@#1}{0pt}%
3536 \csgdef{Xragged@#1}{}%
3537 \csgdef{Xhsizetwocol@#1}{0.45 \hsize}%
3538 \csgdef{Xhsizethreecol@#1}{.3 \hsize}%
3539 \csgdef{Xcolalign@#1}{\raggedright}%
3540 \csgdef{Xnotenumfont@#1}{\normalfont}%
3541 \csgdef{Xnotefontsize@#1}{\footnotesize}%
3542 \csgdef{Xbhooknote@#1}{}%
3543
3544 \csgdef{Xboxlinenum@#1}{0pt}%
3545 \csgdef{Xboxlinenumalign@#1}{L}%
3546
3547 \csgdef{Xboxstartlinenum@#1}{0pt}%
3548 \csgdef{Xboxendlinenum@#1}{0pt}%
3549
3550 \csgdef{Xboxsymlinenum@#1}{0pt}%
3551 \newtoggle{Xnumberonlyfirstinline@#1}%
3552 \newtoggle{Xnumberonlyfirstintwolines@#1}%
3553 \csgdef{Xtwolines@#1}{}%
3554 \csgdef{Xmorethantwolines@#1}{}%
3555 \newtoggle{Xtwolinesbutnotmore@#1}%
3556 \newtoggle{Xtwolinesonlyinsamepage@#1}%
3557 \newtoggle{Xonlypstart@#1}%
3558 \newtoggle{Xpstarteverytime@#1}%
3559 \newtoggle{Xpstart@#1}%
3560 \csgdef{Xsymlinenum@#1}{}%
3561 \newtoggle{Xnonumber@#1}%
3562 \csgdef{Xbeforenumber@#1}{0pt}%
3563 \csgdef{Xafternumber@#1}{0.5em}%
3564 \newtoggle{Xnonbreakableafternumber@#1}%
3565 \csgdef{Xbeforesymlinenum@#1}{\csuse{Xbeforenumber@#1}}%
3566 \csgdef{Xaftersymlinenum@#1}{\csuse{Xafternumber@#1}}%
3567 \csgdef{Xinplaceofnumber@#1}{1em}%
3568 \global\cslet{Xlemmaseparator@#1}{\rbracket}%
3569 \csgdef{Xbeforelemmaseparator@#1}{0em}%

```

```

3570 \csgdef{Xafterlemmaseparator@#1}{0.5em}%
3571 \csgdef{Xinplaceoflemmaseparator@#1}{1em}%
3572 \csgdef{Xbeforenotes@#1}{1.2em \@plus .6em \@minus .6em}%
3573 \csgdef{Xafterrule@#1}{Opt}%
3574 \csgdef{Xtxtbeforenotes@#1}{%
3575 \csgdef{Xmaxhnotes@#1}{0.8\vsizex}%
3576 \newtoggle{Xnoteswidthliketwocolumns@#1}%
3577 \csgdef{Xparafootsep@#1}{%
3578 \csgdef{Xafternote@#1}{1em plus.4em minus.4em}%

```

### XX.3.2 Create inserts, needed to add notes in foot

As regards inserts, see chapter 15 of *The TeXbook* by D. Knuth.

```

3579 \expandafter\newinsert\csname #1footins\endcsname%
3580 \unless\ifnoledgroup%
3581 \expandafter\newinsert\csname mp#1footins\endcsname%
3582 \fi%

```

### XX.3.3 Create commands for critical apparatus, \Afootnote, \Bfootnote etc.

Note the double # in command: it is because command it is made inside another command.

```

3583 \global\newbool{parapparatus@}{\expandafter\newcommand\expandafter *}{\expandafter\newcommand}\csname
3584 \ifnum\@edtext@level>0%
3585 \begingroup%
3586 \newcommand{\content}{##2}%
3587 \ifnumberedpar@%
3588 \ifledRcol%
3589 \ifluatex%
3590 \footnotelang@lua[R]%
3591 \fi%
3592 \@ifundefined{xpg@main@language}%if polyglossia
3593 {}%
3594 {\footnotelang@poly[R]}%
3595 \footnoteoptions@R[##1]{true}%
3596 \xright@appenditem{%
3597 \noexpand\prepare@preXnotes{#1}%
3598 \noexpand\prepare@edindex@fornote{\l@d@nums}%
3599 \unexpanded{\def\sw@list@inedtext}{\expandafter\unexpanded\expandafter{\sw@inthisedtext
3600 \noexpand\csuse{v#1footnote}{#1}%
3601 {\l@d@nums}{\expandonce\@tag}{\expandonce\content}}}%
3602 }\to\inserts@listR
3603 \footnoteoptions@R[##1]{false}%
3604 \global\advance\insert@countR \@ne%
3605 \else%
3606 \ifluatex%
3607 \footnotelang@lua%
3608 \fi%
3609 \@ifundefined{xpg@main@language}%if polyglossia
3610 {}%

```

```

3611         {\footnotelang@poly}%
3612         \footnoteoptions@{##1}{true}%
3613         \xright@appenditem{%
3614             \noexpand\prepare@preXnotes{#1}%
3615             \noexpand\prepare@edindex@for@note{\ld@nums}%
3616         \unexpanded{\def\sw@list@inedtext}{\expandafter\unexpanded\expandafter{\sw@
3617             \noexpand\csuse{v#1footnote}{#1}%
3618             {\ld@nums}{\expandonce\@tag}{\expandonce\content}}}%
3619             }\to\inserts@list
3620             \global\advance\insert@count \@ne%
3621             \footnoteoptions@{##1}{false}%
3622         \fi
3623     \else
3624         \csuse{v#1footnote}{#1}{\{0|0|0|0|0|0|0\}}{##1}%
3625     \fi%
3626 \endgroup%
3627 \else%
3628     \led@err@FootnoteWithoutEdtext%
3629 \fi%
3630 \ignorespaces%
3631 }

```

We need to be able to modify reledmac's footnote macros and restore their

```

3632 \global\csletcs{#1@@footnote}{#1footnote}

```

### XX.3.4 Set standard display

```

3633 \Xarrangement@normal{#1}%

```

End of for critical footnotes.

```

3634 \fi

```

## XX.4 For familiar footnotes

Familiar footnotes are those which end with letters. We look for the `nofamiliar` option of reledmac.

```

3635 \unless\ifnofamiliar@

```

### XX.4.1 Options

```

3636 \newtoggle{parindentX@#1}
3637 \csgdef{hangindentX@#1}{0pt}%
3638 \csgdef{raggedX@#1}{}%
3639 \csgdef{hsizetwocolX@#1}{0.45 \hsize}%
3640 \csgdef{hsizethreecolX@#1}{.3 \hsize}%
3641 \csgdef{colalignX@#1}{\raggedright}%
3642 \csgdef{notenumfontX@#1}{\normalfont}%
3643 \csgdef{notefontsizeX@#1}{\footnotesize}%
3644 \csgdef{bhooknoteX@#1}{}%
3645 \csgdef{afterruleX@#1}{0pt}
3646 \csgdef{beforenotesX@#1}{1.2em \@plus .6em \@minus .6em}

```

```

3647 \csgdef{maxhnotesX@#1}{0.8\vsizex}%
3648 \newtoggle{noteswidthliketwocolumnsX@#1}%
3649 \csgdef{parafootsepX@#1}{}%
3650 \csgdef{afternoteX@#1}{1em plus.4em minus.4em}
3651 % End of for familiar footnotes.
3652 % \subsubsection{Create inserts, needed to add notes in foot}
3653 % As regards inserts, see chapter 15 of the TeXBook by D. Knuth.
3654 % \begin{macrocode}
3655 \expandafter\newinsert\csname footins#1\endcsname%
3656 \unless\ifnoledgroup%
3657 \expandafter\newinsert\csname mpfootins#1\endcsname%
3658 \fi%

```

#### XX.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.

```

3659 \global\expandafter\newcommand\csname footnote#1\endcsname[1]{%
3660 \begingroup%
3661 \prepare@prenotesX{#1}%
3662 \newcommand{\content}{##1}%
3663 \stepcounter{footnote#1}%
3664 \protected@csxdef{@thefnmark#1}{\csuse{thefootnote#1}}%
3665 \nottoggle{nomk@}%Nomk is set to true when using \footnoteXnomk with \parpackage
3666 {\csuse{@footnotemark#1}}%
3667 {}%
3668 \ifluatex%
3669 \xdef\footnote@luatextextdir{\the\luatextextdir}%
3670 \xdef\footnote@luatexpardir{\the\luatexpardir}%
3671 \fi%
3672 \csuse{vfootnote#1}{#1}{\expandonce\content}\m@mmlf@prepare%
3673 \endgroup%
3674 }
3675

```

Then define the counters.

```

3676 \newcounter{footnote#1}
3677 \global\expandafter\renewcommand\csname thefootnote#1\endcsname{\arabic{footnote#1}}

```

Do not forget to initialize series

```

3678 \arrangementX@normal{#1}%
3679 \fi

```

## XX.5 The endnotes

Endnotes are commands like `\Xendnote`, where X is a series letter. First, we check for the `noend` options.

```

3680 \unless\ifnoend@

```

### XX.5.1 The auxiliary file

`\l@d@Xend` Endnotes of all varieties are saved up in a file, one by series, typically named `\jobname`.Xend.  
`\ifl@dend@X` `\l@d@end` is the output stream number for this file, and `\ifl@dend@X` is a flag that is  
`\l@dend@Xtrue` true when the file is open.  
`\l@dend@Xfalse`

```

3681 \expandafter\newwrite\csname l@d@#1end\endcsname%
3682 \expandafter\newif\csname ifl@dend@#1\endcsname%
```

### XX.5.2 The main macro

The `\Xendnote` macro functions to write one endnote to the .Xend 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 does not exceed restrictions on the length of lines in files.

```

3683
3684 \global\expandafter\newcommandx\csname #1endnote\endcsname[2][1,usedefault]{%
3685     \bgroup%
3686     \newlinechar='40%
3687     \global\@noneed@Footnotetrue%
3688     \newcommand{\content}{##2}%
3689     \expandafter\immediate\expandafter\write\csname l@d@#1end\endcsname{%
3690         \expandafter\string\csname #1end\endcsname%
3691         {\ifnumberedpar@l@d@nums\fi}%
3692         {\ifnumberedpar@\expandonce\@tag\fi}%
3693         {\expandonce\content}%
3694         {\#1}%
3695         {\##1}%
3696         \@percentchar%
3697     }%
3698     \egroup%
3699     \ignorespaces%
3700 }
```

`\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.

```

3701
3702 \global\cslet{#1end}{\@gobblefive}
```

We need to store the number of times `\doendnotesbysection` is called for one series.

```

3703 \global\expandafter\newcount\csname #1end@bysection\endcsname%
```

### XX.5.3 The options

```

3704 \csgdef{Xendtwolines@#1}{}%
3705 \csgdef{Xendmorethantwolines@#1}{}%
3706 \newtoggle{Xendtwolinesbutnotmore@#1}{%}
```

```

3707 \newtoggle{Xendtwolinesonlyinsamepage@#1}{}%
3708 \newtoggle{Xendlemmadisablefontselection@#1}%
3709 \csgdef{Xendnotenumfont@#1}{\normalfont}%
3710 \csgdef{Xendnotefontsize@#1}{\footnotesize}%
3711 \csgdef{Xendbhooknote@#1}{}%
3712
3713 \csgdef{Xendboxlinenum@#1}{Opt}%
3714 \csgdef{Xendboxlinenumalign@#1}{L}%
3715
3716 \csgdef{Xendboxstartlinenum@#1}{Opt}%
3717 \csgdef{Xendboxendlinenum@#1}{Opt}%
3718
3719 \csgdef{Xendlemmaseparator@#1}{}%
3720 \csgdef{Xendbeforelemmaseparator@#1}{Oem}%
3721 \csgdef{Xendafterlemmaseparator@#1}{0.5em}%
3722 \csgdef{Xendinplaceoflemmaseparator@#1}{0.5em}%
3723
3724 \newtoggle{Xendparagraph@#1}%
3725 \csgdef{Xendafternote@#1}{1em plus.4em minus.4em}%
3726 \csgdef{Xendsep@#1}{}%
3727
3728 \csgdef{Xendinplaceofnumber@#1}{Opt}%
3729 \newtoggle{Xendnonumber@#1}%

End of endnotes declaration
3730 \fi%

Dump series in \@series
3731 \listxadd{\@series}{#1}
3732 }
3733 }% End of \newseries

```

## XX.6 Init standards series (A,B,C,D,E)

```
3734 \expandafter\newseries\expandafter{\default@series}
```

## XXI Setting series display

### XXI.1 Change series order

`\seriesatbegin` `\seriesatbegin{<s>}` changes the order of series, to put the series `<s>` at the beginning of the list. The series can be the result of a command.

```

3735 \newcommand{\seriesatbegin}[1]{%
3736   \StrDel{\@series}{#1}[\@series]%
3737   \edef\@new{%
3738     \listxadd{\@new}{#1}%
3739     \listxadd{\@new}{\@series}%
3740     \xdef\@series{\@new}%
3741 }

```

`\seriesatend` And `\seriesatend` moves the series to the end of the list.

```

3742 \newcommand{\seriesatend}[1]{%
3743   \StrDel{\@series}{#1}[\@series]%
3744   \edef\@new{%
3745     \listead{\@new}{\@series}%
3746     \listead{\@new}{#1}%
3747     \xdef\@series{\@new}%
3748 }

```

## XXI.2 Test series order

`\ifseriesbefore` `\ifseriesbefore{<seriesA>}{<seriesB>}{<true>}{<false>}` expands `<true>` if `<seriesA>` is printed before `<seriesB>`, expands `<false>` otherwise.

```

3749 \newcommand{\ifseriesbefore}[4]{%
3750   \StrPosition{\@series}{#1}[\@first]%
3751   \StrPosition{\@series}{#2}[\@second]%
3752   \ifnumgreater{\@second}{\@first}{#3}{#4}%
3753 }

```

### XXI.2.1 Get the first series

In some specific case, we need to know the first series of the list of series.

`\@getfirstseries`

```

3754 \newcommand{\@getfirstseries}{%
3755   \ifdefempty{\@series}%
3756   {\xdef\@firstseries{}}%
3757   {\StrChar{\@series}{1}[\@firstseries]}%
3758 }%

```

## XXI.3 Series setting

### XXI.3.1 General way of working

The setting's command (like `\numberonlyfirstinline`), also called “hooks” can be divided in two categories: those which require a string values and those which require a boolean value. The first category includes those which require a length value, because we store the length's expression send by user and we evaluate it only in the commands which requires to know the setting. The second category require boolean value only when it is set to FALSE. Otherwise, we understand the insinuated value is TRUE.

For each “hook” command, we store the value in commands (first category) or a etoolbox's toggle (second category) which names are in the form `<hook>@<series>`. For example when calling `\twolines{<sq.>}`, we store `sq.` in commands `\twolines@A`, `\twolines@B`, `\twolines@C`...for each series defined for use with `reledmac`, or, if the `[<series>]` optional argument was send, for each series of this argument.

These values are tested in some specific places, scattered in all the code, depending of their effects. The default values are defined by the `\newseries@` command.



In order to prevent code duplication, we have created some generic commands. Some of them change the value of any hook send as argument. Some other, getting a hook name, generate the user level commands.

### XXI.3.2 Tools to set options

`\settoggle@series` `\settoggle@series{<series>}{<toggle>}{<value>}` 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 `reload`, reload the footnote setting (call again `\Xarrangement` or `\arrangementX` or ... depending of the footnote display).
- #5 (optional): if not empty, and if #1 is empty, change the hook setting for pseudo-series, as `appref`.

```

3759 \newcommandx{\settoggle@series}[5][4,5,usedefault]{%
3760   \def\do##1{%
3761     \global\settoggle{#2@##1}{#3}%
3762     \ifstrequal{#4}{reload}%
3763     {%
3764       \csuse{Xarrangement@}\csuse{series@display##1}{##1}%
3765       \csuse{arrangementX@}\csuse{series@displayX##1}{##1}%
3766     }%
3767   }%
3768 }%
3769 \ifstreempty{#1}{%
3770   \dolistloop{\@series}%
3771   \ifstreempty{#5}{}%
3772   \docsvlist{#5}%
3773 }
3774 }%
3775 {%
3776   \docsvlist{#1}%
3777 }%
3778 }
```

`\setcommand@series` `\setcommand@series{<series>}{<command>}{<value>}` is a generic command to store hook's value into commands specific to 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 `reload`, reload the footnote setting (call `\footnormal` or `\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 `appref`.

```

3779 \newcommandx{\setcommand@series}[5][4,5,usedefault]{%
3780   \def\do##1{
3781     \csgdef{#2@##1}{#3}
3782     \ifstrequal{#4}{reload}{
3783       \csuse{Xarrangement@\csuse{series@display##1}}{##1}%
3784       \csuse{arrangementX@\csuse{series@displayX##1}}{##1}%
3785     }{}}
3786     \ifstreempty{#1}{%
3787       \dolistloop{@series}%
3788       \ifstreempty{#5}{}%
3789       \docsvlist{#5}
3790     }
3791   }%
3792   {%
3793     \docsvlist{#1}%
3794   }%
3795 }%
```

### XXI.3.3 Tools to generate options commands

`\newhookcommand@series` `\newhookcommand@series\command` `names` is a generic command to add new commands for hooks, like `\Xhsizetwocol`. 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 `\Xtwolines`. The second argument is also used to create commands named `\<hookname><pseudoserries>`, like `\Xtwolinesappref`.

```

3796 \newcommandx{\newhookcommand@series}[2][2,usedefault]{%
3797   \global\expandafter\newcommand\expandafter*\csname #1\endcsname[2][{%
3798     \setcommand@series{##1}{#1}{##2}[][#2]%
3799   }%
3800   \ifstreempty{#2}{}%
3801   \def\do##1{%
3802     \global\expandafter\newcommand\expandafter*\csname #1##1\endcsname[1]{%
3803       \csuse{#1}[##1]{####1}%
3804     }%
3805   }%
3806   \docsvlist{#2}%
3807 }%
3808 }
```

`\newhooktoggle@series` `\newhooktoggle@series\command` `names` is a generic command to add new commands for a new toggle hook, like `\Xnumberonlyfirstinline`. The second argu-

ment is also used to create commands named `\<hookname><pseudoseris>`, like `\Xtwolinesbutnotmoreappref`.

```

3809 \newcommand{\newhooktoggle@series}[2][2,usedefault]{%
3810   \global\expandafter\newcommand\expandafter*\csname #1\endcsname[2][1,2={true},usedefault]{%
3811     \settoggle@series{##1}{#1}{##2}[][#2]%
3812   }%
3813   \ifstrempy{#2}{-}{%
3814     \def\do##1{%
3815       \global\expandafter\newcommand\expandafter*\csname #1##1\endcsname{%
3816         \csuse{#1}[##1]%
3817       }%
3818     }%
3819     \docsvlist{#2}%
3820   }%
3821 }

```

`\newhooktoggle@series@reload` `\newhookcommand@toggle@reload` does the same thing as `\newhooktoggle@series` but the commands created by this macro also reload the series arrangement.

```

3822 \newcommand{\newhooktoggle@series@reload}[1]{%
3823   \global\expandafter\newcommand\expandafter*\csname #1\endcsname[2][1,2={true},usedefault]{%
3824     \settoggle@series{##1}{#1}{##2}[reload]%
3825   }%
3826 }%

```

`\newhookcommand@series@reload` `\newhookcommand@series@reload` does the same thing as `\newhookcommand@series` but the commands created by this macro also reload the series' arrangement.

```

3827 \newcommand{\newhookcommand@series@reload}[1]{%
3828   \global\expandafter\newcommand\expandafter*\csname #1\endcsname[2][]{%
3829     \setcommand@series{##1}{#1}{##2}[reload]%
3830   }%
3831 }

```

### XXI.3.4 Options for critical notes

Before generating the commands that are used to set the critical notes, such as `\Xnumberonlyfirstinline`, `\Xlemmaseparator` and the like, we check the `nocritical` option.

```

3832 \unless\ifnocritical@
3833   \newhooktoggle@series{Xparindent}
3834   \newhookcommand@series{Xtwolines}[appref]
3835   \newhookcommand@series{Xmorethantwolines}[appref]
3836   \newhooktoggle@series{Xtwolinesbutnotmore}[appref]
3837   \newhooktoggle@series{Xtwolinesonlyinsamepage}[appref]
3838   \newhookcommand@series{Xhangindent}
3839   \newhookcommand@series{Xragged}
3840   \newhookcommand@series{Xhsizetwocol}
3841   \newhookcommand@series{Xhsizethreecol}
3842   \newhookcommand@series{Xcolalign}%

```

```

3843 \newhookcommand@series{Xnotenumfont}
3844 \newhookcommand@series{Xbhooknote}
3845 \newhookcommand@series{Xboxsymlinenum}%
3846 \newhookcommand@series{Xsymlinenum}
3847 \newhookcommand@series{Xbeforenumber}
3848 \newhookcommand@series{Xafternumber}
3849 \newhookcommand@series{Xbeforesymlinenum}
3850 \newhookcommand@series{Xaftersymlinenum}
3851 \newhookcommand@series{Xinplaceofnumber}
3852 \newhookcommand@series{Xlemmaseparator}
3853 \newhookcommand@series{Xbeforelemmaseparator}
3854 \newhookcommand@series{Xafterlemmaseparator}
3855 \newhookcommand@series{Xinplaceoflemmaseparator}
3856 \newhookcommand@series{Xtxtbeforenotes}
3857 \newhookcommand@series@reload{Xafterrule}
3858 \newhooktoggle@series{Xnumberonlyfirstinline}
3859 \newhooktoggle@series{Xnumberonlyfirstintwolines}
3860 \newhooktoggle@series{Xnonumber}
3861 \newhooktoggle@series{Xpstart}
3862 \newhooktoggle@series{Xpstarteverytime}%
3863 \newhooktoggle@series{Xonlypstart}
3864 \newhooktoggle@series{Xnonbreakableafternumber}
3865 \newhooktoggle@series{Xlemmadisablefontselection}
3866 \newhookcommand@series@reload{Xmaxhnotes}
3867 \newhookcommand@series@reload{Xbeforenotes}
3868 \newhooktoggle@series@reload{Xnoteswidthliketwocolumns}%
3869 \newhookcommand@series{Xnotefontsize}
3870
3871 \newhookcommand@series{Xboxlinenum}%
3872 \newhookcommand@series{Xboxlinenumalign}%
3873
3874 \newhookcommand@series{Xboxstartlinenum}%
3875 \newhookcommand@series{Xboxendlinenum}%
3876
3877 \newhookcommand@series{Xafternote}%
3878 \newhookcommand@series{Xparafootsep}
3879
3880 \fi

```

### XXI.3.5 Options for familiar notes

Before generating the optional commands for familiar notes, we check the `\nofamiliar` option.

```

3881 \unless\ifnofamiliar@
3882 \newhooktoggle@series{parindentX}
3883 \newhookcommand@series{hangindentX}
3884 \newhookcommand@series{raggedX}
3885 \newhookcommand@series{hsizetwocolX}
3886 \newhookcommand@series{hsizethreecolX}

```

```

3887 \newhookcommand@series{colalignX}%
3888 \newhookcommand@series{notenumfontX}
3889 \newhookcommand@series{bhooknoteX}
3890 \newhookcommand@series@reload{beforenotesX}
3891 \newhookcommand@series@reload{maxhnotesX}
3892 \newhooktoggle@series@reload{noteswidthliketwocolumnsX}%
3893 \newhookcommand@series@reload{afterruleX}
3894 \newhookcommand@series{notefontsizeX}
3895 \newhookcommand@series{afternoteX}
3896 \newhookcommand@series{parafootsepX}
3897 \fi

```

### XXI.3.6 Options for endnotes

Before generating the commands that are used to set the endnotes, such as `\Xnumberonlyfirstinline`, `\Xlemmaseparator+` and the like, we check the `noend` option.

```

3898 \unless\ifnoend@
3899 \newhookcommand@series{Xendtwolines}[apprefwithpage]
3900 \newhookcommand@series{Xendmorethantwolines}[apprefwithpage]
3901 \newhooktoggle@series{Xendtwolinesbutnotmore}[apprefwithpage]
3902 \newhooktoggle@series{Xendtwolinesonlyinsamepage}[apprefwithpage]
3903 \newhookcommand@series{Xendnotenumfont}
3904 \newhookcommand@series{Xendbhooknote}
3905
3906 \newhookcommand@series{Xendboxlinenum}%
3907 \newhookcommand@series{Xendboxlinenumalign}%
3908
3909 \newhookcommand@series{Xendboxstartlinenum}%
3910 \newhookcommand@series{Xendboxendlinenum}%
3911
3912 \newhookcommand@series{Xendnotefontsize}
3913 \newhooktoggle@series{Xendlemmadisablefontselection}
3914 \newhookcommand@series{Xendlemmaseparator}
3915 \newhookcommand@series{Xendbeforelemmaseparator}
3916 \newhookcommand@series{Xendafterlemmaseparator}
3917 \newhookcommand@series{Xendinplaceoflemmaseparator}
3918
3919 \newhooktoggle@series{Xendparagraph}
3920 \newhookcommand@series{Xendafternote}
3921 \newhookcommand@series{Xendsep}
3922
3923 \newhookcommand@series{Xendinplaceofnumber}%
3924 \newhooktoggle@series{Xendnonumber}%
3925 \fi

```

## XXI.4 Hooks for a particular footnote

`\fulllines@` `\fulllines@` toggle is used to print the full lines references, and not the abbreviated form defined by `\Xtwolines` and `\Xmorethantwolines`.

```

3926 \newtoggle{fulllines@}%

\nonum@ \nonum@ toggle is used to disable line number printing in a particular footnote.
3927 \newtoggle{nonum@}

\nosep@ \nonum@ toggle is used to disable the lemma separator in a particular footnote.
3928 \newtoggle{nosep@}

\nomk@ \nomk@ toggle is used by reledpar to remove the footnote mark in the text when using
\footnoteXmk. Read reledpar handbook.
3929 \newtoggle{nomk@}%

```

### XXI.5 Alias

`\Xnolemmaseparator` `\Xnolemmaseparator[<series>]` is just an alias for `\Xlemmaseparator[<series>]{}`.

```

3930 \newcommandx*{\Xnolemmaseparator}[1][1]{\Xlemmaseparator[#1]}

```

## XXII Output routine

Now we begin the output routine and associated things.

### XXII.0.1 Page number management

`\pageno` `\pageno` is a page number, starting at 1, and `\advancepageno` increments the number.

```

\advancepageno 3931 \countdef\pageno=0 \pageno=1
3932 \newcommand*{\advancepageno}{\ifnum\pageno<\z@ \global\advance\pageno\m@ne
3933 \else\global\advance\pageno\@ne\fi}
3934

```

### XXII.0.2 Extra footnotes output

With luck we might only have to change `\@makecol` and `\@reinserts` of the  $\text{\LaTeX}$ 's kernel. Since `reledmac`, we use `etoolbox`'s patching commands instead of overriding. It should provides better compatibility with other package which modify these commands

`\doextrafeet` `\doextrafeet` is the code extending `\@makecol` to cater for the extra `reledmac` feet. We have two categories of extra footnotes. By default, we order the footnote inserts so that the regular footnotes of  $\text{\LaTeX}$  are first, then familiar familiar footnotes and finally the critical footnotes.

```

3935 \newcommand*{\l@ddoxtrafeet}{%
3936 \IfStrEq{familiar-critical}{\@fnpos}
3937 {\do@feetX\Xdo@feet}%
3938 {%
3939 \IfStrEq{critical-familiar}{\@fnpos}%
3940 {\Xdo@feet\do@feetX}%
3941 {\do@feetX\Xdo@feet}%
3942 }%

```

3943 }%  
3944

`\Xdo@feet` `\Xdo@feet` is the code extending `\@makecol` to cater for the extra critical feet.

```
3945 \newcommand*{\Xdo@feet}{%
3946   \setbox\@outputbox \vbox{%
3947     \unvbox\@outputbox
3948     \@opXfeet}}
```

`\@opXfeet` The extra critical feet to be added to the output. The normal way to add one series, `\print@Xnotes` `\print@Xnotes`, is replaced by `reledpar` when using `\Pages`.

```
3949 \newcommand\print@Xnotes[1]{%
3950   \csuse{##1footstart}{##1}%
3951   \csuse{##1footgroup}{##1}%%
3952 }%
```

We print all series of notes by looping on them. We check before printing them that they are not voided.

```
3953 \newcommand*{\@opXfeet}{%
3954   \unless\ifnocritical@%
3955     \gdef\firstXseries@{%
3956       \def\do##1{%
3957         \ifvoid\csuse{##1footins}\else%
3958           \global\skip\csuse{##1footins}=\csuse{Xbeforenotes###1}%
3959           \global\advance\skip\csuse{##1footins} by\csuse{Xafterrule###1}%
3960           \print@Xnotes{##1}%
3961         \fi%
3962       }%
3963     \dolistloop{\@series}%
3964   \fi%
3965 }%
```

`\l@ddodoreinextrafeet` `\l@ddodoreinextrafeet` is the code for catering for the extra footnotes within `\@reinserts`. We use the same category and ordering as in `\l@ddoxtrafeet`.

```
3966 \newcommand*{\l@ddodoreinextrafeet}{%
3967   \IfStrEq{familiar-critical}{\@fnpos}
3968     {\@doreinfeetX\X@doreinfeet}%
3969     {%
3970       \IfStrEq{critical-familiar}{\@fnpos}%
3971       {\X@doreinfeet\@doreinfeetX}%
3972       {\@doreinfeetX\X@doreinfeet}%
3973     }%
3974 }
3975
```

`\X@doreinfeet` `\X@doreinfeet` is the code for catering for the extra critical footnotes within `\@reinserts`.

```
3976 \newcommand*{\X@doreinfeet}{%
3977   \unless\ifnocritical@%
3978   \def\do##1{%
```

```

3979     \ifvoid\csuse{##1footins}\else%
3980         \insert\csuse{##1footins}{\unvbox\csuse{##1footins}}%
3981     \fi}%
3982 \dolistloop{\@series}
3983 \fi%
3984 }
3985
\print@notesX We have to add all the new kinds of familiar footnotes to the output routine. The normal
\do@feetX way to add one series. \print@Xnotes is replaced by reledpar when using \Pages.
\@doreinfeetX
3986 \newcommand\print@notesX[1]{%
3987     \csuse{footstart#1}{#1}%
3988     \csuse{footgroup#1}{#1}%
3989 }%
We print all the series of notes by looping on them. We check before printing them that
they are not voided.
3990 \newcommand*\do@feetX{%
3991     \unless\ifnofamiliar@%
3992         \gdef\firstseriesX@{}%
3993         \setbox\@outputbox \vbox{%
3994             \unvbox\@outputbox%
3995             \def\do##1{%
3996                 \ifvoid\csuse{footins##1}\else%
3997                     \global\skip\csuse{footins##1}=\csuse{beforenotesX@##1}%
3998                     \global\advance\skip\csuse{footins##1} by\csuse{afterruleX@##1}%
3999                     \print@notesX{##1}%
4000                 \fi%
4001             }%
4002             \dolistloop{\@series}}%
4003     \fi%
4004 }%
4005
4006 \newcommand{\@doreinfeetX}{%
4007     \unless\ifnofamiliar@%
4008         \def\do##1{%
4009             \ifvoid\csuse{footins##1}\else
4010                 \insert%
4011                     \csuse{footins##1}
4012                     {\unvbox\csuse{footins##1}}%
4013             \fi%
4014         }%
4015         \dolistloop{\@series}%
4016     \fi%
4017 }%
4018

```



### XXII.0.3 Standard output's commands patching

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.)

```

4019 \ifclassloaded{memoir}{%
    memoir is loaded so we use memoir's built in hooks.
4020   \g@addto@macro{\m@mdoextrafeet}{\l@ddoxtrafeet}%
4021   \g@addto@macro{\m@ddodoreinextrafeet}{\l@ddodoreinextrafeet}%
4022 }{%
    memoir has not been loaded, so patch \@makecol and \@reinserts.
4023   \@ifpackageloaded{fancyhdr}{%
4024     \patchcmd%
4025       {\latex@makecol}%
4026       {\xdef\@freelist{\@freelist\@midlist}}%
4027       {\xdef\@freelist{\@freelist\@midlist}\l@ddoxtrafeet}%
4028       {}%
4029       {\led@error@fail@patch@@@makecol}%
4030   }{%
4031     \patchcmd%
4032       {\@makecol}%
4033       {\xdef\@freelist{\@freelist\@midlist}}%
4034       {\xdef\@freelist{\@freelist\@midlist}\l@ddoxtrafeet}%
4035       {}%
4036       {\led@error@fail@patch@@@makecol}%
4037   }%
4038
4039   \patchcmd%
4040     {\@reinserts}%
4041     {\ifvbox}%
4042     {\l@ddodoreinextrafeet\ifvbox}%
4043     {}%
4044     {\led@error@fail@patch@@@reinserts}%
4045 }
4046

```

It turns out that `\@doclearpage` also needs modifying.

`\if@led@nofoot` We have to check if there are any leftover feet.

```

4047 \newif\if@led@nofoot
4048

```

```

4049 \ifclassloaded{memoir}{%

```

If the memoir class is loaded we hook into its modified `\@doclearpage`.

`\@mem@extranofeet`

```

4050 \g@addto@macro{\@mem@extranofeet}{%%
4051   \def\do#1{%

```

```

4052 \unless\ifnocritical%
4053 \ifvoid\csuse{#1footins}\else\@mem@nofootfalse\fi%
4054 \fi%
4055 \unless\ifnofamiliar%
4056 \ifvoid\csuse{footins#1}\else\@mem@nofootfalse\fi%
4057 \fi%
4058 }
4059 \dolistloop{\@series}%
4060 }%
4061 }{%
```

As memoir is not loaded we have patch \@doclearpage.

```

\@led@testifnofoot
\@doclearpage
4062 \newcommand*{\@led@testifnofoot}{%
4063 \@led@nofoottrue%
4064 \ifvoid\footins\else%
4065 \@led@nofootfalse%
4066 \fi%
4067 \def\do##1{%
4068 \unless\ifnocritical%
4069 \ifvoid\csuse{##1footins}\else%
4070 \@led@nofootfalse%
4071 \fi%
4072 \fi%
4073 \unless\ifnofamiliar%
4074 \ifvoid\csuse{footins##1}\else%
4075 \@led@nofootfalse%
4076 \fi%
4077 \fi%
4078 }%
4079 \dolistloop{\@series}%
4080 }%
4081
4082 \pretocmd%
4083 {\@doclearpage}%
4084 {\@led@testifnofoot}%
4085 {}%
4086 {\led@error@fail@patch@@doclearpage}%
4087
4088 \patchcmd%
4089 {\@doclearpage}%
4090 {\ifvoid\footins}%
4091 {\if\@led@nofoot}%
4092 {}%
4093 {\led@error@fail@patch@@doclearpage}%
4094
4095 }
4096
```

## XXIII Cross referencing

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>}` or `\pstartref`. These reference commands will produce, respectively, the page, line sub-line and pstart 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 `\edlabel{<foo>}`.

`\labelref@list` Set up a new list, `\labelref@list`, to hold the page, line and sub-line numbers for each label.

```
4097 \list@create{\labelref@list}
```

`\zz@@@` A convenience macro to zero two labeling counters in one go.

```
4098 \newcommand*\zz@@@{000|000} % set two counters to zero in one go
4099
```

`\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.<sup>31</sup>

This version of the original `edmac \label` uses `\@bsphack` and `\@esphack` to eliminate extra space problems and also use the  $\TeX$  write methods for the `.aux` file.

Jesse Billett<sup>32</sup> found that the original code could be off by several pages. This version, hopefully cures that, and also allows for non-arabic page numbering.

```
4100 \newcommand*\edlabel}[1]{%
4101   \ifl@ddpairing\ifautopar%
4102     \strut%
4103   \fi\fi%
4104   \@bsphack%
4105   \ifledRcol%
4106     \write\linenum@outR{\string\@lab}%
4107     \ifx\labelref@listR\empty%
4108       \xdef\label@refs{\zz@@@}%
4109     \else%
4110       \gl@p\labelref@listR\to\label@refs%
4111     \fi%
4112     \ifvmode%
4113       \advance\label@refs%
4114     \fi%
```

<sup>31</sup>The remaining macros in this section were kindly revised by Wayne Sullivan, who substantially improved their efficiency and flexibility.

<sup>32</sup>(jdb43@cam.ac.uk) via the ctt thread 'ledmac cross referencing', 25 August 2003.

Use code from the kernel `\label` command to write the correct page number. Also define an `hypertarget` if `hyperref` package is loaded.

```

4115 \protected@write\@auxout{%
4116 {\string\l@dmake@labelsR\space\thepage|\label@refs|\the\c@pstartR|{#1}}%
4117 \ifdef{\hypertarget}{\Hy@raisedlink{\hypertarget{#1}}}{}}%
4118 \else%
4119 \write\linenum@out{\string\@lab}%
4120 \ifx\labelref@list\empty%
4121 \xdef\label@refs{\zz@@}%
4122 \else%
4123 \gl@p\labelref@list\to\label@refs%
4124 \fi%
4125 \ifvmode%
4126 \advancelabel@refs%
4127 \fi%
4128 \protected@write\@auxout{%
4129 {\string\l@dmake@labels\space\thepage|\label@refs|\the\c@pstart|{#1}}%
4130 \ifdef{\hypertarget}{\Hy@raisedlink{\hypertarget{#1}}}{}}%
4131 \fi%
4132 \esphack}%
4133

```

`\advancelabel@refs` In cases where `\edlabel` is the first element in a paragraph, we have a problem with  
`\labelrefsparseline` line counts, because line counts change only at the first horizontal box of the paragraph.  
`\labelrefsparsesubline` 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.

```

4134 \newcounter{line}%
4135 \newcounter{subline}%
4136 \newcommand{\advancelabel@refs}{%
4137 \setcounter{line}{\expandafter\labelrefsparseline\label@refs}%
4138 \stepcounter{line}%
4139 \ifsublines%
4140 \setcounter{subline}{\expandafter\labelrefsparsesubline\label@refs}%
4141 \stepcounter{subline}{1}%
4142 \def\label@refs{\theline|\thesubline}%
4143 \else%
4144 \def\label@refs{\theline|0}%
4145 \fi%
4146 }
4147 \def\labelrefsparseline#1|#2{#1}
4148 \def\labelrefsparsesubline#1|#2{#2}

```

`\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).

#1 page number, #2 line number, #3 sub-line number, #4 pstart number, #5 label.

```

4149 \newcommand*\l@dmake@labels{-}
4150 \def\l@dmake@labels#1|#2|#3|#4|#5{%
4151   \expandafter\ifx\csname the@label#5\endcsname \relax\else
4152     \led@warn@DuplicateLabel{#5}%
4153   \fi
4154   \expandafter\gdef\csname the@label#5\endcsname{#1|#2|#3|#4}%
4155   \ignorespaces}
4156

```

TeX 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.

```

4157 \AtBeginDocument{%
4158   \def\l@dmake@labels#1|#2|#3|#4|#5{-}%
4159 }
4160

```

`\@lab` The `\@lab` command, which appears in the `\linenum@out` file, appends the current values of page, line and sub-line to the `\labelref@list`. These values are defined by the earlier `\@page`, `\@nl`, and the `\sub@on` and `\sub@off` commands appearing in the `\linenum@out` file.

TeX uses the page counter for page numbers. However, it appears that this is not the right place to grab the page number. That task is now done in the `\edlabel` macro. This version of `\@lab` appends just the current line and sub-line numbers to `\labelref@list`.

```

4161
4162 \newcommand*\@lab{-}%
4163   \ifledRcol
4164     \xright@appenditem{\linenumr@p{\line@numR}}{|%
4165       \ifsublines@ \sublinenumr@p{\subline@numR}\else 0\fi}%
4166     \to\labelref@listR
4167   \else
4168     \xright@appenditem{\linenumr@p{\line@num}}{|%
4169       \ifsublines@ \sublinenumr@p{\subline@num}\else 0\fi}%
4170     \to\labelref@list
4171   \fi}

```

`\applabel` `\applabel`, if called in `\edtext` will insert automatically both a start and an end label for the current edtext lines.

```

4172 \newcommand*\applabel[1]{%
4173   \ifnum\@edtext@level>0%
4174     \ifcsundef{the@label#1}{%
4175       \csdef{the@label#1}{applabel}%
4176     }%
4177     {%
4178       \led@warn@DuplicateLabel{#1 (applabel)}%
4179     }%

```

Parse the `\edtext` line numbers.

```
4180 \expandafter\l@dp@rsefootspec\l@d@nums|%
```

Use the  $\TeX$  standard hack for label.

```
4181 \@bsphack%
```

And now, write the data in the auxiliary file.

```
4182 \ifledRcol%
```

```
4183 \protected@write\@auxout{}%
```

```
4184 {\string\l@dmake@labelsR\space\l@dparsedstartpage|\l@dparsedstartline|\l@dparsedsta
```

```
4185 \ifdef{\hypertarget}{\Hy@raisedlink{\hypertarget{#1:start}}{}}{}}%
```

```
4186 \protected@write\@auxout{}%
```

```
4187 {\string\l@dmake@labelsR\space\l@dparsedendpage|\l@dparsedendline|\l@dparsedendsub|
```

```
4188 \else%
```

```
4189 \protected@write\@auxout{}%
```

```
4190 {\string\l@dmake@labels\space\l@dparsedstartpage|\l@dparsedstartline|\l@dparsedstar
```

```
4191 \ifdef{\hypertarget}{\Hy@raisedlink{\hypertarget{#1:start}}{}}{}}%
```

```
4192 \protected@write\@auxout{}%
```

```
4193 {\string\l@dmake@labels\space\l@dparsedendpage|\l@dparsedendline|\l@dparsedendsub|
```

```
4194 \fi%
```

Use the  $\TeX$  standard hack for label.

```
4195 \@esphack%
```

Warning if `\applabel` is called outside of `\edtext`.

```
4196 \else%
```

```
4197 \led@warn@AppLabelOutEdtext{#1}%
```

```
4198 \fi%
```

End of `\applabel`

```
4199 }%
```

`\wrap@edcrossref` `\wrap@edcrossref` is called around all `reledmac` crossref commands, except those which start with `x`. It adds the hyperlink.

```
4200 \newrobustcmd{\wrap@edcrossref}[2]{%
```

```
4201 \ifdef{\hyperlink}%
```

```
4202 {\hyperlink{#1}{#2}}%
```

```
4203 {#2}}%
```

```
4204 }
```

`\edpageref` If the specified label exists, `\edpageref` gives its page number.

`\xpageref` 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 do not 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 `\edlabel` or a normal reference command appears first, or these `x`-commands will always return zeros.

$\TeX$  already defines a `\pageref`, so changing the name to `\edpageref`.

```
4205 \newcommand*{\edpageref}[1]{\l@dref@undefined{#1}\wrap@edcrossref{#1}{\l@dgetref@num{1}{#1}}
```

```
4206 \newcommand*{\xpageref}[1]{\l@dgetref@num{1}{#1}}
```

```
4207 }
```

`\edlineref` If the specified label exists, `\lineref` gives its line number.

```
\xlineref 4208 \newcommand*{\edlineref}[1]{\l@dref@undefined{#1}\wrap@edcrossref{#1}{\l@dgetref@num{2}{#1}}}%
4209 \newcommand*{\xlineref}[1]{\l@dgetref@num{2}{#1}}%
4210
```

`\sublineref` If the specified label exists, `\sublineref` gives its sub-line number.

```
\xsublineref 4211 \newcommand*{\sublineref}[1]{\l@dref@undefined{#1}\wrap@edcrossref{#1}{\l@dgetref@num{3}{#1}}}%
4212 \newcommand*{\xsublineref}[1]{\l@dgetref@num{3}{#1}}%
4213
```

`\pstarteref` If the specified label exists, `\pstarteref` gives its pstart number.

```
\xpstarteref 4214 \newcommand*{\pstarteref}[1]{\l@dref@undefined{#1}\wrap@edcrossref{#1}{\l@dgetref@num{4}{#1}}}%
4215 \newcommand*{\xpstarteref}[1]{\l@dgetref@num{4}{#1}}%
4216
```

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.

`\l@dref@undefined` The `\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.

```
4217 \newcommand*{\l@dref@undefined}[1]{%
4218 \expandafter\ifx\csname the@label#1\endcsname\relax
4219 \led@warn@RefUndefined{#1}%
4220 \fi}
4221
```

`\l@dgetref@num` Next, `\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), sub-line (3) or (4) pstart number. (This switching is done by calling `\l@dlabel@parse`.) The second argument is the label-macro, which because of the `\l@lab` macro above is defined to be a string of the type 123|456|789.

```
4222 \newcommand*{\l@dgetref@num}[2]{%
4223 \expandafter
4224 \ifx\csname the@label#2\endcsname \relax
4225 000%
4226 \else
4227 \expandafter\expandafter\expandafter
4228 \l@dlabel@parse\csname the@label#2\endcsname|#1%
4229 \fi}
4230
```

`\l@dlabel@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@dlabel@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, 3 or 4) which

defines which of the earlier three numbers to pick out. (It was earlier given as the first argument of `\l@dgetref@num`.)

```

4231 \newcommand*{\l@dlabel@parse}{%
4232 \def\l@dlabel@parse#1|#2|#3|#4|#5{%
4233   \ifcase #5%
4234   \or #1%
4235   \or #2%
4236   \or #3%
4237   \or #4%
4238   \fi}

```

`\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 does not, those numbers are set to zero). Then it calls `\linenum` and sets the beginning page, line, and sub-line numbers to those of the place where `\label{mouse}` was placed, and the ending numbers to those at `{elephant}`. The point of this is to be able to manufacture footnote line references to passages which can not be specified in the normal way as the first argument to `\edtext` for one reason or another. Using `\xxref` in the second argument of `\edtext` lets you set things up at least semi-automatically.

```

4239 \newcommand*{\xxref}[2]{%
4240   {%
4241     \expandafter\ifx\csname the@label#1\endcsname \relax%
4242     \expandafter\let\csname the@@label#1\endcsname\zz@@%
4243     \else%
4244     \expandafter\def\csname the@@label#1\endcsname{\l@dgetref@num{1}{#1}|\l@dgetref@num{2}
4245     \fi%
4246     \expandafter\ifx\csname the@label#2\endcsname \relax%
4247     \expandafter\let\csname the@@label#2\endcsname\zz@@%
4248     \else%
4249     \expandafter\def\csname the@@label#2\endcsname{\l@dgetref@num{1}{#2}|\l@dgetref@num{2}
4250     \fi%
4251     \ifdefined\@Rlineflag%
4252     \StrDel{\csuse{the@@label#1}}{\@Rlineflag}[\@tempa]%
4253     \StrDel{\csuse{the@@label#2}}{\@Rlineflag}[\@tempb]%
4254     \else%
4255     \letcs{\@tempa}{the@@label#1}%
4256     \letcs{\@tempb}{the@@label#2}%
4257     \fi%
4258     \linenum{\@tempa|
4259     \@tempb}}}%
4260

```

`\appref` `\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.

`\apprefwithpage` If `\@apprefprefixsingle` is not empty, it prints it before the line number. If `\@apprefprefixmore` is not empty, it prints it before the line numbers when the first line is not the same as the last line. `\apprefwithpage` prints a crossref 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 `\Xtwolinesappref` and `\Xmorethantwolinesappref` are similar to the footnote hooks and `\Xtwolines` `\Xmorethantwolines`.

So, first declare the default value of the hooks for the pseudo-series `appref`. Also declare the internal toggle which are switch by `reledmac`.

```

4261 \xdef\Xtwolines@appref{}%
4262 \xdef\Xmorethantwolines@appref{}%
4263 \newtoggle{Xtwolinesbutnotmore@appref}%
4264 \newtoggle{Xtwolinesonlyinsamepage@appref}%
4265
4266 \xdef\Xendtwolines@apprefwithpage{}%
4267 \xdef\Xendmorethantwolines@apprefwithpage{}%
4268 \newtoggle{Xendtwolinesbutnotmore@apprefwithpage}%
4269 \newtoggle{Xendtwolinesonlyinsamepage@apprefwithpage}%
4270

```

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.

```

4271
4272 \xdef\Xboxstartlinenum@appref{0pt}
4273 \xdef\Xboxendlinenum@appref{0pt}
4274
4275 \xdef\Xendboxstartlinenum@apprefwithpage{0pt}
4276 \xdef\Xendboxendlinenum@apprefwithpage{0pt}
4277

```

Now, declare the default value of `\@apprefprefixsingle` and `\@apprefprefixmore`, and the commands which defines them

```

4278 \newcommand\@apprefprefixsingle{}%
4279 \newcommand\@apprefprefixmore{}%
4280
4281 \newcommand{\apprefprefixsingle}[1]{%
4282   \gdef\@apprefprefixsingle{#1}%
4283 }
4284
4285 \newcommand{\setapprefprefixmore}[1]{%
4286   \gdef\@apprefprefixmore{#1}%
4287 }
4288

```

And now, the main commands: `\appref` and `\apprefwithpage`. These commands call `\printlines` and `\printendlines`. That is why we have previously declared all hooks values tested inside these last commands.

```

4289 \newcommandx{\appref}[2][1,usedefault]{%
4290   \IfStrEq{#1}{fulllines}%
4291   {\toggletrue{fulllines@}}%
4292   {}%
4293   \xdef\@currentseries{appref}%
4294   \ifdefempty{\@apprefprefixmore}%
4295     {\@apprefprefixsingle}%

```

```

4296  {%
4297    \IfEq{\xlineref{#2:start}}{\xlineref{#2:end}}%
4298    {\@apprefprefixsingle}%
4299    {\@apprefprefixmore}%
4300  }%
4301  \printlines\xpageref{#2:start}|\xlineref{#2:start}|\xsublineref{#2:start}|\xpageref{#2:end}%
4302  \togglefalse{fulllines@}%
4303 }%
4304
4305 % \changes{v1.23.0}{2015/05/18}{Debug \cs{Xendtwolines}, \cs{Xendmoreethantwolines}, \cs{Xendmoreethanthreeolines}}
4306 \newcommand{\apprefwithpage}[2][1,usedefault]{%
4307   \IfStrEq{#1}{fulllines}%
4308   {\toggletrue{fulllines@}}%
4309   {}}%
4310 \xdef\currentseries{apprefwithpage}%
4311 \printendlines\xpageref{#2:start}|\xlineref{#2:start}|\xsublineref{#2:start}|\xpageref{#2:end}%
4312 \togglefalse{fulllines@}%
4313 }%

```

`\edmakelabel` Sometimes the `\edlabel` command cannot be used to specify exactly the page and line desired; you can use the `\edmakelabel` macro make your own label. For example, if you say `\edmakelabel{elephant}{10|25|0}` you will have created 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. `\edmakelabel` takes a label, followed by a page and a line number(s) as arguments.  $\TeX$  defines a `\makelabel` macro which is used in lists. Peter Wilson has changed the name to `\edmakelabel`.

```

4314 \newcommand*{\edmakelabel}[2]{\expandafter\xdef\csname the@label#1\endcsname{#2}}
4315

```

(If you are only going to refer to such a label using `\xxref`, then you can omit entries in the same way as with `\linenum` (see VI.3 p. 103 and V.9 p. 77), since `\xxref` makes a call to `\linenum` in order to do its work.)

## XXIV Side notes

Regular `\marginpars` do not work inside numbered text — they do not produce any note but do put an extra unnumbered blank line into the text.

`\@xympar` Changing `\@xympar` a little at least ensures that `\marginpars` in numbered text do not disturb the flow.

```

4316 \pretocmd{\@xympar}%
4317   {\ifnumberedpar@
4318     \led@warn@NoMarginpars
4319     \@esphack
4320   \else}%
4321 {}%
4322 {}%
4323

```

```

4324 \apptocmd{\@xympar}%
4325   {\fi}%
4326   {}
4327   {}
4328

```

We provide side notes as replacement for `\marginpar` in numbered text.

```

\sidenote@margin  These are the sidenote equivalents to \line@margin and \linenummargin for spec-
\sidenotemargin  ifying which margin. The default is the right margin (opposite to the default for line
\l@dgetsidenote@margin  numbers). \l@dgetsidenote@margin returns the number associated to side note mar-
gin:

```

**left** : 0

**right** : 1

**outer** : 2

**inner** : 3

```

4329 \newcount\sidenote@margin
4330 \newcommand*{\sidenotemargin}[1]{%
4331   \l@dgetsidenote@margin{#1}%
4332   \ifnum\@l@dttempcntb>\m@ne
4333     \ifledRcol
4334       \global\sidenote@marginR=\@l@dttempcntb
4335     \else
4336       \global\sidenote@margin=\@l@dttempcntb
4337     \fi
4338   \fi}%
4339 \newcommand*{\l@dgetsidenote@margin}[1]{%
4340   \def\@tempa{#1}\def\@tempb{left}%
4341   \ifx\@tempa\@tempb
4342     \@l@dttempcntb \z@
4343   \else
4344     \def\@tempb{right}%
4345     \ifx\@tempa\@tempb
4346       \@l@dttempcntb \@ne
4347     \else
4348       \def\@tempb{outer}%
4349       \ifx\@tempa\@tempb
4350         \@l@dttempcntb \tw@
4351       \else
4352         \def\@tempb{inner}%
4353         \ifx\@tempa\@tempb
4354           \@l@dttempcntb \thr@@
4355         \else
4356           \led@warn@BadSidenotemargin
4357           \@l@dttempcntb \m@ne
4358         \fi

```

```

4359         \fi
4360     \fi
4361 \fi}
4362 \sidenotemargin{right}
4363

\l@dlp@rbox We need two boxes to store sidenote texts.
\l@drp@rbox 4364 \newbox\l@dlp@rbox
4365 \newbox\l@drp@rbox
4366

\ledlsnotewidth These specify the width of the left/right boxes (initialised to \marginparwidth), their
\ledrsnotewidth distance from the text (initialised to \linenumsep), and the fonts used.
\ledlsnotesep 4367 \newdimen\ledlsnotewidth \ledlsnotewidth=\marginparwidth
\ledrsnotesep 4368 \newdimen\ledrsnotewidth \ledrsnotewidth=\marginparwidth
\ledlsnotefontsetup 4369 \newdimen\ledlsnotesep \ledlsnotesep=\linenumsep
\ledrsnotefontsetup 4370 \newdimen\ledrsnotesep \ledrsnotesep=\linenumsep
4371 \newcommand*{\ledlsnotefontsetup}{\raggedleft\footnotesize}
4372 \newcommand*{\ledrsnotefontsetup}{\raggedright\footnotesize}
4373

\ledleftnote \ledleftnote, \ledrightnote, \ledinnernote, \ledouternote are the user com-
\ledrightnote mands for left, right, inner and outer sidenotes. The two last one are just alias for the
\ledinnernote two first one, depending of the page number. \ledsidenote{<text>} is the command
\ledouternote for a moveable sidenote.
\ledsidenote 4374 \newcommand*{\ledleftnote}[1]{\edtext{}{\l@dlsnote{#1}}}
4375 \newcommand*{\ledrightnote}[1]{\edtext{}{\l@drsnote{#1}}}
4376
4377 \newcommand*{\ledinnernote}[1]{%
4378 \ifodd\c@page% Do not use \page@num, because it is not yet calculated when command is called
4379 \ledleftnote{#1}%
4380 \else%
4381 \ledrightnote{#1}%
4382 \fi%
4383 }
4384
4385 \newcommand*{\ledouternote}[1]{%
4386 \ifodd\c@page% Do not use \page@num, because it is not yet calculated when command is called
4387 \ledrightnote{#1}%
4388 \else%
4389 \ledleftnote{#1}%
4390 \fi%
4391 }
4392
4393 \newcommand*{\ledsidenote}[1]{\edtext{}{\l@dcnote{#1}}}

\l@dlsnote . The ‘footnotes’ for left, right, and moveable sidenotes. The whole scheme is reminiscent
\l@drsnote of the critical footnotes code.
\l@dcnote 4394 \newif\ifrighnoteup

```

```

4395 \rightnoteuptrue
4396
4397 \newcommand*{\l@dlsnote}[1]{%
4398 \begingroup%
4399 \newcommand{\content}{#1}%
4400 \ifnumberedpar@
4401 \ifledRcol%
4402 \xright@appenditem{\noexpand\vl@dlsnote{\expandonce\content}}%
4403 \to\inserts@listR
4404 \global\advance\insert@countR \@ne%
4405 \else%
4406 \xright@appenditem{\noexpand\vl@dlsnote{\expandonce\content}}%
4407 \to\inserts@list
4408 \global\advance\insert@count \@ne%
4409 \fi
4410 \fi\ignorespaces\endgroup}
4411
4412 \newcommand*{\l@drsnote}[1]{%
4413 \begingroup%
4414 \newcommand{\content}{#1}%
4415 \ifnumberedpar@
4416 \ifledRcol%
4417 \xright@appenditem{\noexpand\vl@drsnote{\expandonce\content}}%
4418 \to\inserts@listR
4419 \global\advance\insert@countR \@ne%
4420 \else%
4421 \xright@appenditem{\noexpand\vl@drsnote{\expandonce\content}}%
4422 \to\inserts@list
4423 \global\advance\insert@count \@ne%
4424 \fi
4425 \fi\ignorespaces\endgroup}
4426
4427 \newcommand*{\l@dcsnote}[1]{%
4428 \begingroup%
4429 \newcommand{\content}{#1}%
4430 \ifnumberedpar@
4431 \ifledRcol%
4432 \xright@appenditem{\noexpand\vl@dcsnote{\expandonce\content}}%
4433 \to\inserts@listR
4434 \global\advance\insert@countR \@ne%
4435 \else%
4436 \xright@appenditem{\noexpand\vl@dcsnote{\expandonce\content}}%
4437 \to\inserts@list
4438 \global\advance\insert@count \@ne%
4439 \fi
4440 \fi\ignorespaces\endgroup}
4441

```

\vl@dlsnote Put the left/right text into boxes, but just save the moveable text. \l@dcsnotetext,  
 \vl@drsnote \l@dcsnotetext@l and \l@dcsnotetext@r are etoolbox's lists which will store the  
 \vl@dcsnote

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@dcsootetext in any cases.
- Store the content of \rightsidenote to:
  - \l@dcsootetext if \ledsidenote is to be put on right.
  - \l@dcsootetext@r if \ledsidenote is to be put on left.
- Store the content of \leftsidenote to:
  - \l@dcsootetext if \ledsidenote is to be put on left.
  - \l@dcsootetext@l if \ledsidenote is to be put on right.

```

4442 \newcommand*{\vl@dlsnote}[1]{%
4443   \ifledRcol{%
4444     \l@dttempcntb=\sidenote@marginR%
4445     \ifnum\l@dttempcntb>\@ne%
4446       \advance\l@dttempcntb by\page@numR%
4447     \fi%
4448   \else%
4449     \l@dttempcntb=\sidenote@margin%
4450     \ifnum\l@dttempcntb>\@ne%
4451       \advance\l@dttempcntb by\page@num%
4452     \fi%
4453   \fi%
4454   \ifodd\l@dttempcntb%
4455     \listgadd{\l@dcsootetext@l}{#1}%
4456   \else%
4457     \listgadd{\l@dcsootetext}{#1}%
4458   \fi
4459 }
4460 \newcommand*{\vl@drrsnote}[1]{%
4461   \ifledRcol{%
4462     \l@dttempcntb=\sidenote@marginR%
4463     \ifnum\l@dttempcntb>\@ne%
4464       \advance\l@dttempcntb by\page@numR%
4465     \fi%
4466   \else%
4467     \l@dttempcntb=\sidenote@margin%
4468     \ifnum\l@dttempcntb>\@ne%
4469       \advance\l@dttempcntb by\page@num%
4470     \fi%
4471   \fi%
4472   \ifodd\l@dttempcntb%
4473     \listgadd{\l@dcsootetext}{#1}%
4474   \else%
4475     \listgadd{\l@dcsootetext@r}{#1}%

```

```

4476 \fi%
4477 }
4478 \newcommand*{\vl@dcnote}[1]{\listgadd{\l@dcnotetext}{#1}}
4479

```

`\setl@dlp@rbox` `\setl@dlprbox{<lednums>}{<tag>}{<text>}` puts `<text>` into the `\l@dlp@rbox` box.  
`\setl@drpr@box` And similarly for the right side box. It is these boxes that finally get displayed in the margins.

```

4480 \newcommand*{\setl@dlp@rbox}[1]{%
4481   {\parindent\z@\hspace=\ledlsnotewidth\ledlsnotefontsetup
4482     \global\setbox\l@dlp@rbox
4483     \ifleftnoteup
4484       =\vbox to\z@{\vss #1}%
4485     \else
4486       =\vbox to 0.70\baselineskip{\strut#1\vss}%
4487     \fi}}
4488 \newcommand*{\setl@drp@rbox}[1]{%
4489   {\parindent\z@\hspace=\ledrsnotewidth\ledrsnotefontsetup
4490     \global\setbox\l@drp@rbox
4491     \ifrightrightoteup
4492       =\vbox to\z@{\vss#1}%
4493     \else
4494       =\vbox to0.7\baselineskip{\strut#1\vss}%
4495     \fi}}
4496 \newif\ifleftnoteup
4497 \leftnoteuptrue

```

`\@sidenotesep` This macro is used to separate sidenotes of the same line.

```

4498 \newcommand{\setsidenotesep}[1]{\gdef\@sidenotesep{#1}}
4499 \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 is the result that we put on the sidenote.

```

4500 \newcommand*{\affixside@note}{%
4501   \def\sidenotecontent@{}%
4502   \numgdef{\itemcount@}{0}%
4503   \def\do##1{%
4504     \ifnumequal{\itemcount@}{0}%
4505     {%
4506       \appto\sidenotecontent@{##1}}% Not print not separator before the 1st note
4507     {\appto\sidenotecontent@{\@sidenotesep ##1}%
4508     }%
4509     \numgdef{\itemcount@}{\itemcount@+1}%
4510   }%
4511   \dolistloop{\l@dcnotetext}%
4512   \ifnumgreater{\itemcount@}{1}{\led@err@ManySidenotes}{}%

```

And we do the same for left and right notes (not movable).

```

4513 \gdef\@templ@d{%
4514 \gdef\@templ@n{\l@dcnotesetext\l@dcnotesetext@l\l@dcnotesetext@r}%
4515 \ifx\@templ@d\@templ@n \else%
4516 \if@twocolumn%
4517 \if@firstcolumn%
4518 \setl@dlp@rbox{##1}{\sidenotecontent@}%
4519 \else%
4520 \setl@drp@rbox{\sidenotecontent@}%
4521 \fi%
4522 \else%
4523 \l@dttempcntb=\sidenote@margin%
4524 \ifnum\l@dttempcntb>\@ne%
4525 \advance\l@dttempcntb by\page@num%
4526 \fi%
4527 \ifodd\l@dttempcntb%
4528 \setl@drp@rbox{\sidenotecontent@}%
4529 \gdef\sidenotecontent@{%
4530 \numgdef{\itemcount@}{0}%
4531 \dolistloop{\l@dcnotesetext@l}%
4532 \ifnumgreater{\itemcount@}{1}{\led@err@ManyLeftnotes}{}%
4533 \setl@dlp@rbox{\sidenotecontent@}%
4534 \else%
4535 \setl@dlp@rbox{\sidenotecontent@}%
4536 \gdef\sidenotecontent@{%
4537 \numgdef{\itemcount@}{0}%
4538 \dolistloop{\l@dcnotesetext@r}%
4539 \ifnumgreater{\itemcount@}{1}{\led@err@ManyRightnotes}{}%
4540 \setl@drp@rbox{\sidenotecontent@}%
4541 \fi%
4542 \fi%
4543 \fi%
4544 }
```

## XXV Minipages and such

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 will arrange this so that additional series can be easily added.

```

\l@dfeetbeginmini These will be the hooks in \@iiiminipage and \endminipage.
\l@dfeetendmini   They can be extended to handle other things if necessary.

4545 \ifnoledgroup\else%
4546 \newcommand*{\l@dfeetbeginmini}{\l@dedbeginmini\l@dfambeginmini}
4547 \newcommand*{\l@dfeetendmini}{%
4548 \IfStrEq{critical-familiar}{\@mpfnpos}%
4549 {\l@dedendmini\l@dfamendmini}%

```



```

4550      {%
4551          \IfStrEq{familiar-critical}{\@mpfnpos}%
4552              {\l@dfamendmini\l@dedendmini}%
4553              {\l@dedendmini\l@dfamendmini}%
4554      }%
4555  }%

```

`\l@dedbeginmini` These handle the initiation and closure of critical footnotes in a minipage environment.

```

\l@dedendmini 4556 \newcommand*{\l@dedbeginmini}{%
4557   \unless\ifnocritical@%
4558   \def\do##1{\csletcs{v##1footnote}{mpv##1footnote}}%
4559   \dolistloop{\@series}%
4560   \fi%
4561 }
4562 \newcommand*{\l@dedendmini}{%
4563   \unless\ifnocritical@%
4564   \ifl@dpairing%
4565     \ifledRcol%
4566       \flush@notesR%
4567     \else%
4568       \flush@notes%
4569     \fi%
4570   \fi
4571   \def\do##1{%
4572     \ifvoid\csuse{mp##1footins}\else%
4573     \ifl@dpairing\ifparledgroup%
4574       \ifledRcol%
4575       \dingdef{\parledgroup@beforenotesR}{\parledgroup@beforenotesR+\skip\@nameuse{mp##1footins}}%
4576     \else%
4577     \dingdef{\parledgroup@beforenotesL}{\parledgroup@beforenotesL+\skip\@nameuse{mp##1footins}}%
4578     \fi%
4579     \fi\fi%
4580     \csuse{mp##1footgroup}{##1}%
4581   \fi}%
4582   \dolistloop{\@series}%
4583   \fi%
4584 }%
4585

```

`\l@dfambeginmini` These handle the initiation and closure of familiar footnotes in a minipage environment.

```

\l@dfamendmini 4586 \newcommand*{\l@dfambeginmini}{%
4587   \unless\ifnofamiliar@%
4588   \def\do##1{\csletcs{vfootnote##1}{mpvfootnote##1}}%
4589   \dolistloop{\@series}%
4590   \fi%
4591 }%
4592
4593 \newcommand*{\l@dfamendmini}{%
4594   \unless\ifnofamiliar@%
4595   \def\do##1{%

```

```

4596     \ifvoid\csuse{mpfootins##1}\else%
4597     \csuse{mpfootgroup##1}{##1}%
4598     \fi}%
4599     \dolistloop{\@series}%
4600 \fi%
4601 }%

```

`\@iiiminipage` This is our extended form of the kernel `\@iiiminipage` defined in `ltboxes.dtx`.

```

4602 \patchcmd%
4603 {\@iiiminipage}%
4604 {\let\@footnotetext\@mpfootnotetext}%
4605 {\let\@footnotetext\@mpfootnotetext\l@dfetbeginmini}%
4606 {}%
4607 {\led@error@fail@patch@\@iiiminipage}%

```

`\endminipage` This is our extended form of the kernel `\endminipage` defined in `ltboxes.dtx`.

```

4608 \patchcmd%
4609 {\endminipage}%
4610 {\footnoterule}%
4611 {\footnoterule\l@advance@parledgroup@beforenormalnotes}%
4612 {}%
4613 {\led@error@fail@patch@endminipage}%
4614
4615 \patchcmd%
4616 {\endminipage}%
4617 {\@minipagefalse}%
4618 {\l@dfetendmini\@minipagefalse}%
4619 {}%
4620 {\led@error@fail@patch@endminipage}%
4621

```

`\l@dunboxmpfoot` `\l@dunboxmpfoot` insert normal footnotes for `ledgroup`.

`\l@advance@parledgroup@beforenormalnotes` 4622 `\newcommand*{\l@dunboxmpfoot}{%`

```

4623     \vskip\skip\@mpfootins
4624     \normalcolor
4625     \footnoterule
4626     \l@advance@parledgroup@beforenormalnotes
4627     \unvbox\@mpfootins%
4628 }

```

When using parallel `ledgroup`, we need to store the vertical space added before footnote, in order to compensate them between left and right pages.

```

4629 \newcommand{\l@advance@parledgroup@beforenormalnotes}{%
4630   \ifparledgroup
4631     \ifl@dpairing
4632     \ifledRcol
4633     \dingdef{\parledgroup@beforenotesR}{\parledgroup@beforenotesR+\skip\@mpfootins}
4634     \else
4635     \dingdef{\parledgroup@beforenotesL}{\parledgroup@beforenotesL+\skip\@mpfootins}
4636     \fi

```

```

4637     \fi
4638 \fi
4639 }

```

**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.

```

4640 \newenvironment{ledgroup}{%
4641   \resetprevpage@num%
4642   \def\@mpfn{mpfootnote}\def\thempfn{\thempfootnote}\c@mpfootnote\z@%
4643   \let\@footnotetext\@mpfootnotetext
4644   \l@dfetbeginmini%
4645 }{%
4646   \par
4647   \unskip
4648   \ifvoid\@mpfootins\else
4649     \l@dunboxmpfoot
4650   \fi
4651   \l@dfetendmini%
4652 }
4653

```

**ledgroupsize** `\begin{ledgroupsize}[\langle pos \rangle]{\langle width \rangle}`

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  $\langle width \rangle$  minipage. The optional  $\langle pos \rangle$  controls the sideways position of numbered text.

```

4654 \newenvironment{ledgroupsize}[2][1]{%

```

Set the various text measures.

```

4655   \hsize #2\relax
4656   %% \textwidth #2\relax
4657   %% \columnwidth #2\relax

```

Initialize fills for centering.

```

4658   \let\ledllfill\hfil
4659   \let\ledrlfill\hfil
4660   \def\@tempa{#1}\def\@tempb{1}%

```

Left adjusted numbered lines

```

4661     \ifx\@tempa\@tempb
4662       \let\ledllfill\relax
4663     \else
4664       \def\@tempb{r}%
4665       \ifx\@tempa\@tempb

```

Right adjusted numbered lines

```

4666       \let\ledrlfill\relax
4667     \fi
4668   \fi

```

Set up the footnoting.

```

4669   \def\@mpfn{mpfootnote}\def\thempfn{\thempfootnote}\c@mpfootnote\z@

```

```

4670 \let\@footnotetext\@mpfootnotetext
4671 \l@dfetbeginmini%
4672 }{%
4673 \par
4674 \unskip
4675 \ifvoid\@mpfootins\else
4676 \l@dunboxmpfoot
4677 \fi
4678 \l@dfetendmini%
4679 }
4680

```

Close the \ifnoledgroup@\else.

```

4681 \fi%

```

\ifledgroupnotesL@ These boolean tests check if we are in the notes of a ledgroup. If we are, we do not  
 \ifledgroupnotesR@ number the lines. It could be useful for parallel ledgroup of reledpar.

```

4682 \newif\ifledgroupnotesL@
4683 \newif\ifledgroupnotesR@

```

## XXVI Indexing

Here is some code for indexing using page and line numbers.

First, ensure that imakeidx or indextools is loaded *before* eledmac.

```

4684 \AtBeginDocument{%
4685 \unless\ifl@imakeidx%
4686 \@ifpackageloaded{imakeidx}{\led@error@ImakeidxAfterEledmac}{}%
4687 \fi%
4688 \unless\ifl@indextools%
4689 \@ifpackageloaded{indextools}{\led@error@indextoolsAfterEledmac}{}%
4690 \fi%
4691 }

```

\pagelinesep In order to get a correct line number we have to use the label/ref mechanism. These  
 \edindexlab macros are for that.

```

\c@labidx 4692 \newcommand{\pagelinesep}{-}
4693 \newcommand{\edindexlab}{${}}
4694 \newcounter{labidx}
4695 \setcounter{labidx}{0}
4696

```

\doedindexlabel This macro sets an \edlabel.

```

4697 \newcommand{\doedindexlabel}{%
4698 \stepcounter{labidx}%
4699 \edlabel{\edindexlab\thelabidx}%
4700 }
4701

```

`\thepageline` This macro makes up the page/line number combo from the label/ref.

```
4702 \newcommand{\thepageline}{%
4703   \thepage%
4704   \pagelinesep%
4705   \xlineref{\edindexlab\thelabidx}%
4706 }
```

`\thestartpageline` These macros make up the page/line start/end number when the `\edindex` command  
`\theendpageline` is called in critical notes.

```
4707 \newcommand{\thestartpageline}{%
4708   \l@dparsedstartpage%
4709   \pagelinesep%
4710   \l@dparsedstartline%
4711 }
4712 \newcommand{\theendpageline}{%
4713   \l@dparsedendpage%
4714   \pagelinesep%
4715   \l@dparsedendline%
4716 }
```

`\if@edindex@fornote@true` This boolean test is switching at the beginning of each critical note, to allow index referring to this note.

```
4717 \newif\if@edindex@fornote@
```

`\prepare@edindex@fornote` This macro is called at the beginning of each critical note. It switches some parameters, to allow index referring to this note, with reference to page and line number. It also defines `\@ledinnote@command` which will be printed as an encapsulating command after the |.

```
4718 \newcommand{\prepare@edindex@fornote}[1]{%
4719   \l@dp@rsefootspec#1|}%
4720   \@edindex@fornote@true%
4721 }
```

`\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.

```
4722 \newcommand{\get@edindex@ledinnote@command}{%
4723   \ifxindy%
4724     \gdef\@ledinnote@command{%
4725       ledinnote\thelabidx%
4726     }%
4727     \ifxindyhyperref%
4728       \immediate\write\eledmac@xindy@out{%
4729         (define-attributes ("ledinnote\thelabidx"))^^J
4730         \space\space(markup-locref^^J
4731         \eledmacmarkuplocrefdepth^^J
4732         :open "\string\ledinnote[\edindexlab\thelabidx]{\@index@command}{^^J
```

```

4733         :close "}"^^J
4734         :attr "ledinnote\thelabidx"^^J
4735     )
4736 }%
4737 \else%
4738     \immediate\write\eledmac@xindy@out{%
4739         (define-attributes ("ledinnote\thelabidx"))^^J
4740         \space\space(markup-locref^^J
4741         \eledmacmarkuplocrefdepth^^J
4742         :open "\string\ledinnote{\@index@command}"^^J
4743         :close "}"^^J
4744         :attr "ledinnote\thelabidx"^^J
4745     )
4746 }%
4747 \fi%

```

If we do not use xindy option, \@ledinnote@command will produce something like ledinnote{formattingcommand}.

```

4748 \else%
4749     \gdef\@ledinnote@command{%
4750         ledinnote[\edindexlab\thelabidx]{\@index@command}%
4751     }%
4752 \fi%
4753 }

```

`\get@index@command` This macro is used to analyse if a text to be indexed has a command after a |.

```

4754 \def\get@index@command#1|#2+{%
4755     \gdef\@index@txt{#1}%
4756     \gdef\@index@command{#2}%
4757     \xdef\@index@parenthesis{}%
4758     \IfBeginWith{\@index@command}{(}{%
4759         \StrGobbleLeft{\@index@command}{1}[\@index@command@]%
4760         \global\let\@index@command\@index@command@%
4761         \xdef\@index@parenthesis{(%}%
4762     }{%
4763     \IfBeginWith{\@index@command}{)}{%
4764         \StrGobbleLeft{\@index@command}{1}[\@index@command@]%
4765         \global\let\@index@command\@index@command@%
4766         \xdef\@index@parenthesis{)%}%
4767     }{%
4768 }

```

`\ledinnote` 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.

`\ledinnotehyperpage`  
`\ledinnotemark`

```

4769 \newcommand{\ledinnote}[3][1,usedefault]{%
4770     \ifboolexpr{%
4771         test{\ifdefequal{\iftrue}{\ifHy@hyperindex}}%
4772         or%
4773         bool {xindyhyperref@}%

```

```

4774 }%
4775 {%
4776 \csuse{#2}{\hyperlink{#1}{\ledinnotemark{#3}}}%
4777 }%
4778 {%
4779 \csuse{#2}{\ledinnotemark{#3}}%
4780 }%
4781 }%
4782 \newcommand{\ledinnotehyperpage}[2]{\csuse{#1}{\ledinnotemark{\hyperpage{#2}}}}%
4783 \newcommand{\ledinnotemark}[1]{#1\emph{n}}%

```

Eledmac and ledmac were using the specific indexing tools of the memoir in order to allow multiple index. However, eledmac used imakeidx or indextools tools when one these two package was loaded. This system forced to maintained a double code, which was not very useful. Since reledmac, we use only the imakeidx or indextools tools.

The memoir class provides more flexible indexing than the standard classes. We need different code if the memoir class is being used, except if imakeidx or indextools is used.

```

\edindex Write the index information to the idx file.
\@wredindex 4784 \newcommand{\@wredindex}[2][1=\expandonce\jobname,usedefault]{%#1 = the index name, #2 = the text
4785 \global\let\old@Rlineflag\@Rlineflag%
4786 \gdef\@Rlineflag{}%
4787 \ifl@imakeidx%
4788 \if@edindex@fornote%
4789 \IfSubStr[1]{#2}{|}{\get@index@command#2+}{\get@index@command#2|+}%
4790 \get@edindex@ledinnote@command%
4791 \expandafter\imki@wrindexentry{#1}{\@index@txt|(\@ledinnote@command){\thestartpageline}%
4792 \expandafter\imki@wrindexentry{#1}{\@index@txt|)\@ledinnote@command}{\theendpageline}%
4793 \else%
4794 \get@edindex@hyperref{#2}%
4795 \imki@wrindexentry{#1}{\@index@txt\@edindex@hyperref}{\thepageline}%
4796 \fi%
4797 \else%
4798 \if@edindex@fornote%
4799 \IfSubStr[1]{#2}{|}{\get@index@command#2+}{\get@index@command#2|+}%
4800 \get@edindex@ledinnote@command%
4801 \expandafter\protected@write\@indexfile{}%
4802 {\string\indexentry{\@index@txt|(\@ledinnote@command){\thestartpageline}
4803 }%
4804 \expandafter\protected@write\@indexfile{}%
4805 {\string\indexentry{\@index@txt|)\@ledinnote@command}{\theendpageline}
4806 }%
4807 \else%
4808 \protected@write\@indexfile{}%
4809 {\string\indexentry{#2}{\thepageline}
4810 }%
4811 \fi%
4812 \fi%
4813 \endgroup
4814 \global\let\@Rlineflag\old@Rlineflag%

```

```
4815 \@esphack%
4816 }
```

Need to add the definition of `\edindex` to `\makeindex`, and initialise `\edindex` to do nothing.

```
4817 \pretocmd{\makeindex}{%
4818   \def\edindex{\@bsphack
4819     \doedindexlabel
4820     \begingroup
4821     \@sanitize
4822     \@wredindex}}{}{}
4823 \newcommand{\edindex}[1]{\@bsphack\@esphack}
```

`\hyperlinkformat` `\hyperlinkformat` command is to be used to have both a internal hyperlink and a format, when indexing.

```
4824 \newcommand{\hyperlinkformat}[3]{%
4825   \ifstrempy{#1}%
4826     {\hyperlink{#2}{#3}}%
4827     {\csuse{#1}{\hyperlink{#2}{#3}}}%
4828 }
```

`\hyperlinkR` `\hyperlinkR` command is to be used to create a internal hyperlink and `\ledRflag`, when indexing.

```
4829 \newcommand{\hyperlinkR}[2]{%
4830   \hyperlink{#1}{#2\@Rlineflag}%
4831 }%
4832
```

`\hyperlinkformatR` `\hyperlinkformatR` command is to be used to create a internal hyperlink, a format and a `\@Rlineflag`, when indexing.

```
4833 \newcommand{\hyperlinkformatR}[3]{%
4834   \hyperlinkformat{#1}{#2}{#3\@Rlineflag}%
4835 }%
4836
```

`\get@edindex@hyperref` `\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`).

```
4837 \newcommand{\get@edindex@hyperref}[1]{%
```

We have to disable temporary spaces to work through a xstring bug (or feature?)

```
4838   \edef\temp@{%
4839     \catcode`\ =9 %space need for catcode
4840     #1%
4841     \catcode`\ =10 % space need for catcode
4842   }%
```

Now, we define `\@edindex@hyperref` if the hyperindex of `hyperref` is enabled.

```
4843   \ifdefequal{\iftrue}{\ifHy@hyperindex}{%
4844     \IfSubStr{\temp@}{|}%
4845     {\get@index@command#1+%}
```



```

4846 \ifledRcol%
4847 \gdef\@edindex@hyperref{|\@index@parenthesis %space kept
4848 hyperlinkformatR{\@index@command}%
4849 {\edindexlab\thelabidx}}%
4850 \else%
4851 \gdef\@edindex@hyperref{|\@index@parenthesis %space kept
4852 hyperlinkformat{\@index@command}%
4853 {\edindexlab\thelabidx}}%
4854 \fi%
4855 }%
4856 {\get@index@command#1|+%
4857 \ifledRcol%
4858 \gdef\@edindex@hyperref{|hyperlinkR{\edindexlab\thelabidx}}%
4859 \else%
4860 \gdef\@edindex@hyperref{|hyperlink{\edindexlab\thelabidx}}%
4861 \fi%
4862 }%
4863 }%

4864 % If we use both xindy and hyperref, first get the \cs{index@command} command.
4865 % Then define \cs{@edindex@hyperref} in the form \verb+eledmacXXX+
4866 % \begin{macrocode}
4867 {\ifxindyhyperref%
4868 \IfSubStr{\temp@}{|}%
4869 {\get@index@command#1|+%
4870 {\get@index@command#1|+%
4871 \gdef\@edindex@hyperref{|eledmac\thelabidx}%

```

If we start a reference range by a opening parenthesis, store the \thelabidx for the current \edindex, then define \@edindex@hyperref in the form |(eledmac\thelabidx.

```

4872 \IfStrEq{\@index@parenthesis}{(}%
4873 {%
4874 \csxdef{xindyparenthesis@\@index@txt}{\thelabidx}%
4875 \gdef\@edindex@hyperref{|(eledmac\thelabidx}%
4876 }%
4877 {}%

```

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 eledmacXXX where XXX is the \thelabidx of the opening \edindex.

```

4878 \IfStrEq{\@index@parenthesis}{)}%
4879 {%
4880 \xdef\@edindex@hyperref{)|eledmac\csuse{xindyparenthesis@\@index@txt}}%
4881 \global\csundef{xindyparenthesis@\@index@txt}%
4882 }%

```

Write in the .xdy file the attributes of the location.

```

4883 {%
4884 \immediate\write\eledmac@xindy@out{%
4885 (define-attributes ("eledmac\thelabidx"))^^J

```

```

4886      \space\space(markup-locoref^^J
4887      \eledmacmarkuplocorefdepth^^J
4888      :open "\string\hyperlink%
4889      \ifledRcol R\fi%
4890      {\edindexlab\thelabidx}%
4891      {\ifdefempty{\@index@command}%
4892      {}%
4893      {\@backslashchar\@index@command}%
4894      {"^^J
4895      :close "}}"^^J
4896      :attr "eledmac\thelabidx"^^J
4897      )
4898      }%
4899      }%

```

And now, in any other case.

```

4900      \else%
4901      \gdef\@index@txt{#1}%
4902      \gdef\@edindex@hyperref{}%
4903      \fi%
4904      }%
4905 }

```

## XXVII Verse

The original code is principally Wayne Sullivan's code from *edstanza*. However, the code has been many time modified by Maïeul Rouquette in order to obtain new features and improved compatibility with *reledpar*.

### XXVII.1 Hanging symbol management

`\@hangingsymbol` The macro `\@hangingsymbol` is used to insert a symbol on each hanging of verses. It is set by user level macro `\sethangingsymbol`.  
`\sethangingsymbol`  
`\ifinstanza` For example, in french typographie the symbol is '['. We obtain it by the next code:

```
\sethangingsymbol{[,}
```

The `\ifinstanza` boolean is used to be sure that we are in a stanza part.

```

4906 \def\@hangingsymbol{}
4907 \newcommand*{\sethangingsymbol}[1]{%
4908   \gdef\@hangingsymbol{#1}%
4909 }%
4910 \newif\ifinstanza

```

`\inserthangingsymbol` The boolean `\inserthangingsymbol` is set to TRUE when `\@lock` is greater than 1,  
`\ifinserthangingsymbol` i.e. when we are not in the first line of a verse. The switch of `\inserthangingsymbol` is made in `\do@line` before the printing of line but after the line number calculation.

```

4911 \newif\ifinserthangingsymbol
4912 \newcommand{\inserthangingsymbol}{%
4913 \ifinserthangingsymbol%
4914   \ifinstanza%
4915     \@hangingsymbol%
4916   \fi%
4917 \fi%
4918 }

```

## XXVII.2 Using & character

`\ampersand` Within a stanza the `\&` 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.

```

4919 \newcommand*{\ampersand}{\char`\&}
4920

```

## XXVII.3 Code category setting

`\stanza@count` Before we can define the main macros we need to save and reset some category codes.  
`\stanzaindentbase` To save the current values we use `\next` and `\body` from the `\loop` macro.

```

4921 \chardef\body=\catcode`\@
4922 \catcode`\@=11
4923 \chardef\next=\catcode`\&
4924 \catcode`\&=\active
4925

```

## XXVII.4 Stanza count and indent

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 `\stanzaindentbase` is 20pt.

```

4926 \newcount\stanza@count
4927 \newlength{\stanzaindentbase}
4928 \setlength{\stanzaindentbase}{20pt}
4929

```

`\strip@szacnt` The indentations of stanza lines are non-negative integer multiples of the unit called  
`\setstanzavalues` `\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.

```

4930 \def\strip@szacnt#1,#2|{\def\@tempb{#1}\def\@tempa{#2|}}
4931 \newcommand*{\setstanzavalues}[2]{\def\@tempa{#2,,|}%
4932   \stanza@count\z@
4933   \def\next{\expandafter\strip@szacnt\@tempa

```

```

4934         \ifx\@tempb\empty\let\next\relax\else
4935         \expandafter\mathchardef\csname #1@\number\stanza@count
4936         @\endcsname\@tempb\relax
4937         \advance\stanza@count\@ne\fi\next}%
4938     \next}
4939

```

`\setstanzaindents` In the original edmac, `\setstanzavalues{sza}{\langle... \rangle}` had to be called to set the in-  
`\setstanzapenalties` dents, and similarly `\setstanzavalues{szp}{\langle... \rangle}`. to set the penalties. `\setstanzaindents`  
and `\setstanzapenalties` macros are a convenience to give the user one less thing  
to worry about (misspelling the first argument).

```

4940 \newcommand*\setstanzaindents[1]{\setstanzavalues{sza}{#1}}
4941 \newcommand*\setstanzapenalties[1]{\setstanzavalues{szp}{#1}}
4942 %

```

`\managestanza@modulo` Since version 0.13, the `stanzaindentsrepetition` counter can be used when the in-  
dentation 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.

```

4943 \newcounter{stanzaindentsrepetition}
4944 \newcount\stanza@modulo
4945
4946 \newcommand*\managestanza@modulo[0]{
4947     \advance\stanza@modulo\@ne
4948     \ifnum\stanza@modulo>\value{stanzaindentsrepetition}
4949         \stanza@modulo\@ne
4950     \fi
4951 }

```

`\stanzaindent` The macro `\stanzaindent`, when called at the beginning of a verse, changes the in-  
`\stanzaindent*` dentation normally defined for this verse by `\setstanzaindent`. The starred version  
skips the current verse for the repetition of stanza indent.

```

4952 \newcommand{\stanzaindent}[1]{%
4953     \hspace{\dimexpr#1\stanzaindentbase-\parindent\relax}%
4954     \ignorespaces%
4955 }%
4956 \WithSuffix\newcommand\stanzaindent*[1]{%
4957     \stanzaindent{#1}%
4958     \global\advance\stanza@modulo-\@ne%
4959     \ifnum\stanza@modulo=0%
4960         \global\stanza@modulo=\value{stanzaindentsrepetition}%
4961     \fi%
4962     \ignorespaces%
4963 }%

```

## XXVII.5 Main work

`\stanza@line` 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.

```

4964 \newcommandx{\stanza@line}[1][1]{
4965   \ifnum\value{stanzaindentrepetition}=0
4966     \parindent=\csname sza@number\stanza@count
4967       \@endcsname\stanzaindentbase
4968   \else
4969     \parindent=\csname sza@number\stanza@modulo
4970       \@endcsname\stanzaindentbase
4971     \managestanza@modulo
4972   \fi
4973   \pstart[#1]\stanza@hang\ignorespaces}
4974 \xdef\stanza@hang{\noexpand\leavevmode\noexpand\startlock
4975   \hangindent\expandafter
4976   \noexpand\csname sza@0@\endcsname\stanzaindentbase
4977   \hangafter\@ne}
4978 \def\sza@penalty{\count@\csname szp@number\stanza@count \@endcsname
4979   \ifnum\count@>\@M\advance\count@-\@M\penalty-\else
4980   \penalty\fi\count@}

```

`\@startstanza` 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 `\&`.

```

4981 \xdef\@startstanza[#1]{%
4982   \noexpand\instanzatrue\expandafter
4983   \begingroup%
4984   \catcode`\noexpand\&\active%
4985   \global\stanza@count\@ne\stanza@modulo\@ne
4986   \noexpand\ifnum\expandafter\noexpand
4987   \csname sza@0@\endcsname=\z@\let\noexpand\stanza@hang\relax
4988   \let\noexpand\endlock\relax\noexpand\else\interlinepenalty
4989   \@M\rightskip\z@ plus 1fil\relax\noexpand\fi\noexpand\ifnum
4990   \expandafter\noexpand\csname szp@0@\endcsname=\z@
4991   \let\noexpand\sza@penalty\relax\noexpand\fi%
4992   \def\noexpand&{%

```

```

4993      \noexpand\newverse[] []}%
4994      \def\noexpand\&\noexpand\@stopstanza}%
4995      \noexpand\stanza@line[#1]}
4996
4997 \newcommandx{\stanza}[1][1,usedefault]{\@startstanza[#1]}
4998
4999 \newcommandx{\@stopstanza}[1][1,usedefault]{%
5000   \unskip%
5001   \endlock%
5002   \pend[#1]%
5003   \endgroup%
5004   \instanzafalse%
5005 }
5006
5007 \newcommandx*{\newverse}[2][1,2,usedefault]{%
5008   \unskip%
5009   \endlock\pend[#1]\sza@penalty\global%
5010   \advance\stanza@count\@ne\stanza@line[#2]%
5011   }
5012

```

`\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`.

```

5013 \newcommand*{\flagstanza}[2][\stanzaindentbase]{%
5014   \hskip -#1\llap{#2}\hskip #1\ignorespaces}
5015

```

## XXVII.6 Restore catcode and penalties

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 `\stanza` `\&`. 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.

```

5016 \catcode`\&=\next
5017 \catcode`\@=\body
5018 \setstanzavalues{szp}{0}
5019

```

## XXVIII Arrays and tables

### XXVIII.1 Preamble: macro as environment

The following is borrowed, and renamed, from the `amsmath` package. See also the CTT thread ‘`eq` 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.

```
5020 \newtoks\@emptytoks
5021
```

The rest is from `amsmath`.

`\l@denbody` A token register to contain the body.

```
5022 \newtoks\l@denbody
5023
```

`\addtol@denbody` `\addtol@denbody{arg}` adds `arg` to the token register `\l@denbody`.

```
5024 \newcommand{\addtol@denbody}[1]{%
5025   \global\l@denbody\expandafter{\the\l@denbody#1}}
5026
```

`\l@dcollect@body` The macro `\l@dcollect@body` starts the scan for the `\end{env}` 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}` would call `\l@dcollect@body\cenv`.

```
5027 \newcommand{\l@dcollect@body}[1]{%
5028   \l@denbody{\expandafter#1\expandafter{\the\l@denbody}}%
5029   \edef\processl@denbody{\the\l@denbody\noexpand\end{\@currenvir}}%
5030   \l@denbody\@emptytoks \def\l@dbegin@stack{b}%
5031   \begingroup
5032     \expandafter\let\csname\@currenvir\endcsname\l@dcollect@@body
5033     \edef\processl@denbody{\expandafter\noexpand\csname\@currenvir\endcsname}%
5034     \processl@denbody%
5035   }%
5036
```

`\l@dpush@begins` When adding a piece of the current environment's contents to `\l@denbody`, we scan it to check for additional `\begin` tokens, and add a 'b' to the stack for any that we find.

```
5037 \def\l@dpush@begins#1\begin#2{%
5038   \ifx\end#2\else b\expandafter\l@dpush@begins\fi}
5039
```

`\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.

```
5040 \def\l@dcollect@@body#1\end#2{%
5041   \edef\l@dbegin@stack{\l@dpush@begins#1\begin\end
```

```

5042             \expandafter\@gobble\l@dbegin@stack}%
5043 \ifx\@empty\l@dbegin@stack
5044   \endgroup
5045   \@checkend{#2}%
5046   \addtol@denbody{#1}%
5047 \else
5048   \addtol@denbody{#1\end{#2}}%
5049 \fi
5050 \processl@denbody % A little tricky! Note the grouping
5051 }
5052

```

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}[1]{\colorbox{red}{#1}}
>
> \makeatletter
> \newenvironment{redbox}{\collect@body \redbox}{\end{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.

```

\documentclass{article}
\usepackage{color}
\usepackage{amsmath}

```



```

\newcommand{\redbox}[1]{\colorbox{red}{#1}}
\makeatletter
\newenvironment{coloredbox}[1]{%
  \def\next@{\colorbox{#1}}%
  \collect@body\next@
}{%

% ignore spaces at begin and end of environment
\newenvironment{coloredboxII}[1]{%
  \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}%
}
\def\@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>
```

## XXVIII.2 Tabular environments

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. Peter Wilson have made some cosmetic changes to the original code and reimplemented some things so they are more LaTeX-like. All the commentary are from Peter Wilson, as are any mistake or errors.

However, Maïeul Rouquette has modified code in order to add new features of `eledmac` and `reledmac`.

### XXVIII.2.1 Disabling and restoring commands

```
\l@dtabnoexpands More no expansion for critical and familiar footnotes in tabular environment.

5053 \newcommand*{\l@dtabnoexpands}{%
5054   \let\rtab=0%
5055   \let\ctab=0%
5056   \let\ltab=0%
5057   \let\rtabtext=0%
5058   \let\ltabtext=0%
5059   \let\ctabtext=0%
5060   \let\edbeforetab=0%
5061   \let\edaftertab=0%
5062   \let\edatleft=0%
5063   \let\edatright=0%
5064   \let\edvertline=0%
5065   \let\edvertdots=0%
5066   \let\edrowfill=0%
5067 }
5068

\disable@familiarnotes Macros to disable and restore familiar notes, to prevent them from printing multiple
\restore@familiarnotes times in edtabularx and edarrayx environments.

5069 \newcommand{\disable@familiarnotes}{%
5070   \unless\ifnofamiliar{%
5071     \def\do##1{%
5072       \csletcs{footnote@##1}{footnote##1}%
5073       \expandafter\renewcommand \csname footnote##1\endcsname[1]{%
5074         \protected@csxdef{@thefnmark##1}{\csuse{thefootnote##1}}%
5075         \csuse{@footnotemark##1}%
5076       }%
5077     }%
```

```

5078   \dolistloop{\@series}%
5079   \fi%
5080 }%
5081 \newcommand{\restore@familiarnotes}{%
5082   \unless\ifnofamiliar%
5083     \def\do##1{%
5084       \csletcs{footnote##1}{footnote@@##1}%
5085     }%
5086   \dolistloop{\@series}%
5087   \fi%
5088 }%
5089

```

`\disable@sidenotes` The same, for side notes.

```

\restore@sidenotes 5090 \newcommand{\disable@sidenotes}{%
5091   \let\@@ledrightnote\ledrightnote%
5092   \let\@@ledleftnote\ledleftnote%
5093   \let\@@ledsidenote\ledsidenote%
5094   \let\ledrightnote@gobble%
5095   \let\ledleftnote@gobble%
5096   \let\ledsidenote@gobble%
5097 }%
5098 \newcommand{\restore@sidenotes}{%
5099   \let\ledrightnote\@@ledrightnote%
5100   \let\ledleftnote\@@ledleftnote%
5101   \let\ledsidenote\@@ledsidenote%
5102 }%

```

`\disable@notes` Disable/restore side and familiar notes.

```

\restore@notes 5103 \newcommand{\disable@notes}{%
5104   \disable@sidenotes%
5105   \disable@familiarnotes%
5106 }%
5107 \newcommand{\restore@notes}{%
5108   \restore@sidenotes%
5109   \restore@familiarnotes%
5110 }%

```

`\EDTEXT` We need to be able to modify the `\edtext` macros and also restore their original definitions.

```

5111 \let\EDTEXT=\edtext
5112 \newcommand{\xedtext}[2]{\EDTEXT{#1}{#2}}

```

`\EDLABEL` We need to be able to modify and restore the `\edlabel` macro.

```

\xedlabel 5113 \let\EDLABEL=\edlabel
5114 \newcommand*\xedlabel[1]{\EDLABEL{#1}}

```

`\EDINDEX` Macros supporting modification and restoration of `\edindex`.

```

\xedindex 5115 \let\EDINDEX=\edindex
\nulledindex

```

```

5116 \newcommand{\xedindex}{\@bsphack%
5117   \@ifnextchar [{\l@d@index}{\l@d@index[\jobname]}}
5118 \newcommand{\nulledindex}[2][\jobname]{\@bsphack\@esphack}
5119

\@line@num Macro supporting restoration of \linenum.
5120 \let\@line@num=\linenum

\l@dgobblearg \l@dgobbleoptarg[\arg]{\arg} replaces these two arguments (first is optional) by
\relax.
5121 \newcommand*\l@dgobbleoptarg[2][\relax]%
5122

\Relax
\NEXT 5123 \let\Relax=\relax
5124 \let\NEXT=\next
5125

\l@dmodforedtext Modify and restore various macros for when \edtext is used.
\l@drestoreforedtext 5126 \newcommand{\l@dmodforedtext}{%
5127   \let\edtext\relax
5128   \def\do##1{\global\csletcs{##1footnote}{\l@dgobbleoptarg}}%
5129   \dolistloop{\@series}%
5130   \let\edindex\nulledindex
5131   \let\linenum@gobble}
5132 \newcommand{\l@drestoreforedtext}{%
5133   \def\do##1{\global\csletcs{##1footnote}{##1@footnote}}
5134   \dolistloop{\@series}%
5135   \let\edindex\xedindex}

\l@dnullfills Nullify and restore some column fillers, etc.
\l@drestorefills 5136 \newcommand{\l@dnullfills}{%
5137   \def\edlabel##1{}%
5138   \def\edrowfill##1##2##3{}%
5139 }
5140 \newcommand{\l@drestorefills}{%
5141   \def\edrowfill##1##2##3{\@EDROWFILL@{##1}{##2}{##3}}%
5142 }
5143

\letsforverteilen Gathers some lets and other code that is common to the *verteilen* macros.
5144 \newcommand{\letsforverteilen}{%
5145   \let\edtext\xedtext
5146   \let\edindex\xedindex
5147   \def\do##1{\global\csletcs{##1footnote}{##1@footnote}}
5148   \dolistloop{\@series}%
5149   \let\linenum\@line@num
5150   \hilfsskip=\l@dcowidth%
5151   \advance\hilfsskip by -\wd\hilfsbox

```

```

5152 \def\edlabel##1{\xedlabel{##1}}
5153

```

`\disablel@dtabfeet` Declarations for using or using `\edtext` inside tabulars. The default at this point is for  
`\enablel@dtabfeet` `\edtext`.

```

5154 \newcommand\disablel@dtabfeet{\l@modforedtext}%
5155 \newcommand\enablel@dtabfeet{\l@drestoreforedtext}%

```

### XXVIII.2.2 Counters, boxes and lengths

`\l@dampcount` `\l@dampcount` is a counter for the & column dividers and `\l@dcolcount` is a counter  
`\l@dcolcount` for the columns.

```

5156 \newcount\l@dampcount
5157 \l@dampcount=1\relax
5158 \newcount\l@dcolcount
5159 \l@dcolcount=0\relax
5160

```

`\hilfsbox` Some (temporary) helper items.

```

\hilfsskip 5161 \newbox\hilfsbox
\Hilfsbox 5162 \newskip\hilfsskip
\hilfscount 5163 \newbox\Hilfsbox
5164 \newcount\hilfscount
5165

```

30 columns should be adequate (compared to the original 60). These are the column widths. (Originally these were German spelled numbers e.g., `\eins`, `\zwei`, etc).

```

5166 \newdimen\dcoli
5167 \newdimen\dcolii
5168 \newdimen\dcoliii
5169 \newdimen\dcoliv
5170 \newdimen\dcolv
5171 \newdimen\dcolvi
5172 \newdimen\dcolvii
5173 \newdimen\dcolviii
5174 \newdimen\dcolix
5175 \newdimen\dcolx
5176 \newdimen\dcolxi
5177 \newdimen\dcolxii
5178 \newdimen\dcolxiii
5179 \newdimen\dcolxiv
5180 \newdimen\dcolxv
5181 \newdimen\dcolxvi
5182 \newdimen\dcolxvii
5183 \newdimen\dcolxviii
5184 \newdimen\dcolxix
5185 \newdimen\dcolxx
5186 \newdimen\dcolxxi
5187 \newdimen\dcolxxii

```

```

5188 \newdimen\dcollxiii
5189 \newdimen\dcollxiv
5190 \newdimen\dcollxv
5191 \newdimen\dcollxvi
5192 \newdimen\dcollxvii
5193 \newdimen\dcollxviii
5194 \newdimen\dcollxix
5195 \newdimen\dcollxxx
5196 \newdimen\dcollerr % added for error handling
5197

```

`\l@dcollwidth` This is a cunning way of storing the columnwidths indexed by the column number `\l@dcollcount`, like an array. (was `\Dimenzuordnung`)

```

5198 \newcommand{\l@dcollwidth}{\ifcase \the\l@dcollcount \dcoli %???
5199 \or \dcoli \or \dcolii \or \dcoliii
5200 \or \dcoliv \or \dcolv \or \dcolvi
5201 \or \dcolvii \or \dcolviii \or \dcolix \or \dcolx
5202 \or \dcolxi \or \dcolxii \or \dcolxiii
5203 \or \dcolxiv \or \dcolxv \or \dcolxvi
5204 \or \dcolxvii \or \dcolxviii \or \dcolxix \or \dcolxx
5205 \or \dcolxxi \or \dcolxxii \or \dcolxxiii
5206 \or \dcolxxiv \or \dcolxxv \or \dcolxxvi
5207 \or \dcolxxvii \or \dcolxxviii \or \dcolxxix \or \dcolxxx
5208 \else \dcollerr \fi}
5209

```

`\step1@dcollcount` This increments the column counter, and issues an error message if it is too large.

```

5210 \newcommand*{\step1@dcollcount}{\advance\l@dcollcount\@ne
5211 \ifnum\l@dcollcount>30\relax
5212 \led@err@TooManyColumns
5213 \fi}
5214

```

`\l@dsetmaxcolwidth` Sets the column width to the maximum value seen so far.

```

5215 \newcommand{\l@dsetmaxcolwidth}{%
5216 \ifdim\l@dcollwidth < \wd\hilfsbox
5217 \l@dcollwidth = \wd\hilfsbox
5218 \else \relax \fi}
5219

```

`\measuremcell` Measure (recursively) the width required for a math cell.

```

5220 \def\measuremcell #1{%
5221 \ifx #1\ \ifnum\l@dcollcount=0\let\NEXT\relax%
5222 \else\l@dcheckcols%
5223 \l@dcollcount=0%
5224 \let\NEXT\measuremcell%
5225 \fi%
5226 \else\setbox\hilfsbox=\hbox{$\displaystyle{#1}$}%
5227 \step1@dcollcount%

```

```

5228      \l@dsetmaxcolwidth%
5229      \let\NEXT\measuremcell%
5230      \fi\NEXT}
5231

```

`\measuretcell` Measure (recursively) the width required for a text cell.

```

5232 \def\measuretcell #1{%
5233   \ifx #1\ \ifnum\l@dcolcount=0\let\NEXT\relax%
5234           \else\l@dcheckcols%
5235           \l@dcolcount=0%
5236           \let\NEXT\measuretcell%
5237           \fi%
5238   \else\setbox\hilfsbox=\hbox{#1}%
5239   \step\l@dcolcount%
5240   \l@dsetmaxcolwidth%
5241   \let\NEXT\measuretcell%
5242   \fi\NEXT}
5243

```

`\measuremrow` Measure (recursively) the width required for a math row.

```

5244 \def\measuremrow #1\{%
5245   \ifx #1&\let\NEXT\relax%
5246   \else\measuremcell #1&\&\&%
5247   \let\NEXT\measuremrow%
5248   \fi\NEXT}

```

`\measuretrow` Measure (recursively) the width required for a text row.

```

5249 \def\measuretrow #1\{%
5250   \ifx #1&\let\NEXT\relax%
5251   \else\measuretcell #1&\&\&%
5252   \let\NEXT\measuretrow%
5253   \fi\NEXT}
5254

```

`\edtabcolsep` The length `\edtabcolsep` controls the distance between columns.

```

5255 \newskip\edtabcolsep
5256 \global\edtabcolsep=10pt
5257

```

`\variab`

```

5258 \newcommand{\variab}{\relax}
5259

```

`\l@dcheckcols` Check that the number of columns is consistent.

```

5260 \newcommand*{\l@dcheckcols}{%
5261   \ifnum\l@dcolcount=1\relax
5262   \else
5263   \ifnum\l@dampcount=1\relax
5264   \else

```

```

5265     \ifnum\l@dcolcount=\l@dampcount\relax
5266     \else
5267       \l@d@err@UnequalColumns
5268     \fi
5269   \fi
5270   \l@dampcount=\l@dcolcount
5271 \fi}
5272

```

`\edfilldimen` A length.

```

5273 \newdimen\edfilldimen
5274 \edfilldimen=0pt
5275

```

`\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 `\dcol`.

```

5276 \newcounter{addcolcount}
5277 \renewcommand{\theadcolcount}{\roman{addcolcount}}

```

### XXVIII.2.3 Tabular typesetting

`\setmcellright` Typeset (recursively) cells of display math right justified.

```

5278 \def\setmcellright #1&{\def\edlabel##1{}}%
5279     \let\edindex\nulledindex
5280     \ifx #1\\ \ifnum\l@dcolcount=0%\removelastskip
5281         \let\Next\relax%
5282     \else\l@dcolcount=0%
5283         \let\Next=\setmcellright%
5284     \fi%
5285 \else%
5286     \disablel@dtabfeet%
5287     \step1@dcolcount%
5288     \disable@notes%
5289     \setbox\hilfsbox=\hbox{$\displaystyle{#1}$}%
5290     \restore@notes%
5291     \letsforverteilen%
5292     \hskip\hilfsskip$\displaystyle{#1}$%
5293     \hskip\edtabcolsep%
5294     \let\Next=\setmcellright%
5295 \fi\Next}
5296

```

`\settcellright` Typeset (recursively) cells of text right justified.

```

5297 \def\settcellright #1&{\def\edlabel##1{}}%
5298     \let\edindex\nulledindex
5299     \ifx #1\\ \ifnum\l@dcolcount=0%\removelastskip
5300         \let\Next\relax%
5301     \else\l@dcolcount=0%
5302         \let\Next=\settcellright%

```



```

5303         \fi%
5304     \else%
5305         \disablel@dtabfeet%
5306         \stepl@dc colcount%
5307         \disable@notes%
5308         \setbox\hilfsbox=\hbox{#1}%
5309         \restore@notes%
5310         \letsforverteilen%
5311         \hskip\hilfsskip#1%
5312         \hskip\edtabcolsep%
5313         \let\Next=\settcellright%
5314     \fi\Next}

```

`\setmcellleft` Typeset (recursively) cells of display math left justified.

```

5315 \def\setmcellleft #1&{\def\edlabel##1{}%
5316     \let\edindex\nulledindex
5317     \ifx #1\ \ifnum\l@dc colcount=0 \let\Next\relax%
5318         \else\l@dc colcount=0%
5319             \let\Next=\setmcellleft%
5320         \fi%
5321     \else \disablel@dtabfeet%
5322         \stepl@dc colcount%
5323         \disable@notes%
5324         \setbox\hilfsbox=\hbox{$\displaystyle{#1}$}%
5325         \restore@notes%
5326         \letsforverteilen%
5327         $\displaystyle{#1}$\hskip\hilfsskip\hskip\edtabcolsep%
5328         \let\Next=\setmcellleft%
5329     \fi\Next}
5330

```

`\settcellleft` Typeset (recursively) cells of text left justified.

```

5331 \def\settcellleft #1&{\def\edlabel##1{}%
5332     \let\edindex\nulledindex
5333     \ifx #1\ \ifnum\l@dc colcount=0 \let\Next\relax%
5334         \else\l@dc colcount=0%
5335             \let\Next=\settcellleft%
5336         \fi%
5337     \else \disablel@dtabfeet%
5338         \stepl@dc colcount%
5339         \disable@notes%
5340         \setbox\hilfsbox=\hbox{#1}%
5341         \restore@notes%
5342         \letsforverteilen%
5343         #1\hskip\hilfsskip\hskip\edtabcolsep%
5344         \let\Next=\settcellleft%
5345     \fi\Next}

```

`\setmcellcenter` Typeset (recursively) cells of display math centered.

```

5346 \def\setmcellcenter #1&{\def\edlabel##1{}}%
5347   \let\edindex\nulledindex
5348   \ifx #1\\ \ifnum\l@dc@count=0\let\Next\relax%
5349     \else\l@dc@count=0%
5350       \let\Next=\setmcellcenter%
5351     \fi%
5352   \else \disablel@dt@feet%
5353     \step1@dc@count%
5354     \disable@notes%
5355     \setbox\hilfsbox=\hbox{$\displaystyle{#1}$}%
5356     \restore@notes%
5357     \letsforverteilen%
5358     \hskip 0.5\hilfsskip$\displaystyle{#1}$\hskip0.5\hilfsskip%
5359     \hskip\edtabcolsep%
5360     \let\Next=\setmcellcenter%
5361   \fi\Next}
5362

```

`\settcclcenter` Typeset (recursively) cells of text centered.

```

5363 \def\settcclcenter #1&{\def\edlabel##1{}}%
5364   \let\edindex\nulledindex
5365   \ifx #1\\ \ifnum\l@dc@count=0 \let\Next\relax%
5366     \else\l@dc@count=0%
5367       \let\Next=\settcclcenter%
5368     \fi%
5369   \else \disablel@dt@feet%
5370     \step1@dc@count%
5371     \disable@notes%
5372     \setbox\hilfsbox=\hbox{#1}%
5373     \restore@notes%
5374     \letsforverteilen%
5375     \hskip 0.5\hilfsskip #1\hskip 0.5\hilfsskip%
5376     \hskip\edtabcolsep%
5377     \let\Next=\settcclcenter%
5378   \fi\Next}
5379

```

`\NEXT`

```

5380 \let\NEXT=\relax
5381

```

`\setmrowright` Typeset (recursively) rows of right justified math.

```

5382 \def\setmrowright #1\\{%
5383   \ifx #1& \let\NEXT\relax
5384   \else \centerline{\setmcellright #1&\\&\\&}
5385   \let\NEXT=\setmrowright
5386   \fi\NEXT}

```

`\settroright` Typeset (recursively) rows of right justified text.

```

5387 \def\settroright #1\{%
5388   \ifx #1& \let\NEXT\relax
5389   \else \centerline{\settcellright #1&\&\&}
5390         \let\NEXT=\settroright
5391   \fi\NEXT}
5392

```

`\setmrowleft` Typeset (recursively) rows of left justified math.

```

5393 \def\setmrowleft #1\{%
5394   \ifx #1& \let\NEXT\relax
5395   \else \centerline{\setmcellleft #1&\&\&\&}
5396         \let\NEXT=\setmrowleft
5397   \fi\NEXT}

```

`\settrorleft` Typeset (recursively) rows of left justified text.

```

5398 \def\settrorleft #1\{%
5399   \ifx #1& \let\NEXT\relax
5400   \else \centerline{\settcellleft #1&\&\&\&}
5401         \let\NEXT=\settrorleft
5402   \fi\NEXT}
5403

```

`\setmrowcenter` Typeset (recursively) rows of centered math.

```

5404 \def\setmrowcenter #1\{%
5405   \ifx #1& \let\NEXT\relax%
5406   \else \centerline{\setmcellcenter #1&\&\&\&}
5407         \let\NEXT=\setmrowcenter
5408   \fi\NEXT}

```

`\settrorcenter` Typeset (recursively) rows of centered text.

```

5409 \def\settrorcenter #1\{%
5410   \ifx #1& \let\NEXT\relax
5411   \else \centerline{\settcellcenter #1&\&\&\&}
5412         \let\NEXT=\settrorcenter
5413   \fi\NEXT}
5414

```

`\nullsetzen`

```

5415 \newcommand{\nullsetzen}{%
5416   \step1@dcolcount%
5417   \l@dcolwidth=0pt%
5418   \ifnum\l@dcolcount=30\let\NEXT\relax%
5419         \l@dcolcount=0\relax
5420   \else\let\NEXT\nullsetzen%
5421   \fi\NEXT}
5422

```

`\edatleft` `\edatleft[ $\langle math \rangle\{\langle symbol \rangle\}\langle len \rangle]$` . Left  $\langle symbol \rangle$ ,  $2\langle len \rangle$  high with prepended  $\langle math \rangle$  vertically centered.

```

5423 \newcommand{\edatleft}[3][\@empty]{%
5424   \ifx#1\@empty
5425     \vbox to 10pt{\vss\hbox{$\left#2\vrule width0pt height #3
5426               depth 0pt \right. $\hss}\vfil}
5427   \else
5428     \vbox to 4pt{\vss\hbox{$#1\left#2\vrule width0pt height #3
5429               depth 0pt \right. $\}\vfil}
5430   \fi}

\edatright \edatright[ $]{\langle symbol \rangle}{\langle len \rangle}. Right \langle symbol \rangle, 2\langle len \rangle high with appended
\langle math \rangle vertically centered.

5431 \newcommand{\edatright}[3][\@empty]{%
5432   \ifx#1\@empty
5433     \vbox to 10pt{\vss\hbox{$\left.\vrule width0pt height #3
5434               depth 0pt \right#2 $\hss}\vfil}
5435   \else
5436     \vbox to 4pt{\vss\hbox{$\left.\vrule width0pt height #3
5437               depth 0pt \right#2 #1 $\}\vfil}
5438   \fi}
5439

\edvertline \edvertline{ $len$ } vertical line  $len$  high.

5440 \newcommand{\edvertline}[1]{\vbox to 8pt{\vss\hbox{\vrule height #1}\vfil}}
5441

\edvertdots \edvertdots{ $len$ } vertical dotted line  $len$  high.

5442 \newcommand{\edvertdots}[1]{\vbox to 1pt{\vss\vbox to #1%
5443   {\cleaders\hbox{$\math@th\hbox{.}\vbox to 0.5em{ }$}\vfil}}}
5444

\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@.

5445 \newcommand{\l@dtabaddcols}[2]{%
5446   \l@dccheckstartend{#1}{#2}%
5447   \ifl@dstartendok
5448     \setcounter{addcolcount}{#1}%
5449     \@whilenum \value{addcolcount}<#2\relax \do
5450     {\advance\edfilldimen by \the\csname dcol\theaddcolcount\endcsname
5451     \advance\edfilldimen by \edtabcolsep
5452     \stepcounter{addcolcount}}%
5453     \advance\edfilldimen by \the\csname dcol\theaddcolcount\endcsname
5454   \fi
5455 }
5456

\ifl@dstartendok \l@dccheckstartend{ $startcol$ }{ $endcol$ } checks that the values of  $startcol$  and
\l@dccheckstartend  $endcol$  are sensible. If they are then \ifl@dstartendok is set TRUE, otherwise it
is set FALSE.

5457 \newif\ifl@dstartendok$ 
```

```

5458 \newcommand{\l@dccheckstartend}[2]{%
5459   \l@dstartendoktrue
5460   \ifnum #1<\@ne
5461     \l@dstartendokfalse
5462     \led@err@LowStartColumn
5463   \fi
5464   \ifnum #2>30\relax
5465     \l@dstartendokfalse
5466     \led@err@HighEndColumn
5467   \fi
5468   \ifnum #1>#2\relax
5469     \l@dstartendokfalse
5470     \led@err@ReverseColumns
5471   \fi
5472 }
5473

```

`\edrowfill` `\edrowfill{<startcol>}{<endcol>}` fill fills columns `<startcol>` to `<endcol>` inclusive with `<fill>` (e.g. `\hrulefill`, `\upbracefill`). This is a  $\text{\LaTeX}$  style reimplementation and generalization of the original `\waklam`, `\Waklam`, `\waklamec`, `\wastricht` and `\wapunktel` macros.

```

5474 \newcommand*{\edrowfill}[3]{%
5475   \l@dtabaddcols{#1}{#2}%
5476   \hb@xt@ \the\l@dcwidth{\hb@xt@ \the\edfilldimen{#3}\hss}}
5477 \let\@edrowfill=\edrowfill
5478 \def\@EDROWFILL@#1#2#3{\@edrowfill@{#1}{#2}{#3}}
5479

```

`\edbeforetab` The macro `\edbeforetab{<text>}{<math>}` puts `<text>` at the left margin before array cell entry `<math>`. Conversely, the macro `\edaftertab{<math>}{<text>}` puts `<text>` at the right margin after array cell entry `<math>`. `\edbeforetab` should be in the first column and `\edaftertab` in the last column. The following macros support these.

`\leftltab` `\leftltab{<text>}` for `\edbeforetab` in `\ltab`.

```

5480 \newcommand{\leftltab}[1]{%
5481   \hb@xt@ \z@{\vbox{\edtabindent%
5482     \moveleft\Hilfsskip\hbox{\ #1}}\hss}}
5483

```

`\leftrtab` `\leftrtab{<text>}{<math>}` for `\edbeforetab` in `\rtab`.

```

5484 \newcommand{\leftrtab}[2]{%
5485   #2\hb@xt@ \z@{\vbox{\edtabindent%
5486     \advance\Hilfsskip by\dcoli%
5487     \moveleft\Hilfsskip\hbox{\ #1}}\hss}}
5488

```

`\leftctab` `\leftctab{<text>}{<math>}` for `\edbeforetab` in `\ctab`.

```

5489 \newcommand{\leftctab}[2]{%
5490   \hb@xt@ \z@{\vbox{\edtabindent\l@dcwidth=\l@dampcount%

```

```

5491 \advance\Hilfsskip by 0.5\dcoli%
5492 \setbox\hilfsbox=\hbox{\def\edlabel##1}%
5493 \disablel@dtabfeet$\displaystyle{#2}$}%
5494 \advance\Hilfsskip by -0.5\wd\hilfsbox%
5495 \moveleft\Hilfsskip\hbox{\ #1}}\hss}%
5496 #2}
5497

```

`\rightctab` `\rightctab{<math>}<text>}` for `\edaftertab` in `\ctab`.

```

5498 \newcommand{\rightctab}[2]{%
5499 \setbox\hilfsbox=\hbox{\def\edlabel##1}%
5500 \disablel@dtabfeet#2}\l@dampcount=\l@dcolcount%
5501 #1\hb@xt@#2\z@\vbox{\edtabindent\l@dcolcount=\l@dampcount%
5502 \advance\Hilfsskip by 0.5\l@dcolwidth%
5503 \advance\Hilfsskip by -\wd\hilfsbox%
5504 \setbox\hilfsbox=\hbox{\def\edlabel##1}%
5505 \disablel@dtabfeet$\displaystyle{#1}$}%
5506 \advance\Hilfsskip by -0.5\wd\hilfsbox%
5507 \advance\Hilfsskip by \edtabcolsep%
5508 \moveright\Hilfsskip\hbox{ #2}}\hss}%
5509 }
5510

```

`\rightltab` `\rightltab{<math>}<text>}` for `\edaftertab` in `\ltab`.

```

5511 \newcommand{\rightltab}[2]{%
5512 \setbox\hilfsbox=\hbox{\def\edlabel##1}%
5513 \disablel@dtabfeet#2}\l@dampcount=\l@dcolcount%
5514 #1\hb@xt@#2\z@\vbox{\edtabindent\l@dcolcount=\l@dampcount%
5515 \advance\Hilfsskip by\l@dcolwidth%
5516 \advance\Hilfsskip by-\wd\hilfsbox%
5517 \setbox\hilfsbox=\hbox{\def\edlabel##1}%
5518 \disablel@dtabfeet$\displaystyle{#1}$}%
5519 \advance\Hilfsskip by-\wd\hilfsbox%
5520 \advance\Hilfsskip by\edtabcolsep%
5521 \moveright\Hilfsskip\hbox{ #2}}\hss}%
5522 }
5523

```

`\rightrtab` `\rightrtab{<math>}<text>}` for `\edaftertab` in `\rtab`.

```

5524 \newcommand{\rightrtab}[2]{%
5525 \setbox\hilfsbox=\hbox{\def\edlabel##1}%
5526 \disablel@dtabfeet#2}%
5527 #1\hb@xt@#2\z@\vbox{\edtabindent%
5528 \advance\Hilfsskip by-\wd\hilfsbox%
5529 \advance\Hilfsskip by\edtabcolsep%
5530 \moveright\Hilfsskip\hbox{ #2}}\hss}%
5531 }
5532

```

`\rtab` `\rtab{<body>}` typesets `<body>` as an array with the entries right justified.

`\edbeforetab`

`\edaftertab`

The process is first to measure the  $\langle body \rangle$  to get the column widths, and then in a second pass to typeset the body.

```

5533 \newcommand{\rtab}[1]{%
5534   \l@dnnullfills
5535   \def\edbeforetab##1##2{\lefttab{##1}{##2}}%
5536   \def\edaftertab##1##2{\righttab{##1}{##2}}%
5537   \measurebody{#1}%
5538   \l@drestorefills
5539   \variab
5540   \setmrowright #1\\&\\%
5541   \enablel@dtabfeet}
5542
```

`\measurebody` `\measurebody{ $\langle body \rangle$ }` measures the array  $\langle body \rangle$ .

```

5543 \newcommand{\measurebody}[1]{%
5544   \disablel@dtabfeet%
5545   \l@dcolcount=0%
5546   \nullsetzen%
5547   \l@dcolcount=0
5548   \measuremrow #1\\&\\%
5549   \global\l@dampcount=1}
5550
```

`\rtabtext` `\rtabtext{ $\langle body \rangle$ }` typesets  $\langle body \rangle$  as a tabular with the entries right justified.

```

5551 \newcommand{\rtabtext}[1]{%
5552   \l@dnnullfills
5553   \measuretbody{#1}%
5554   \l@drestorefills
5555   \variab
5556   \settroright #1\\&\\%
5557   \enablel@dtabfeet}
5558
```

`\measuretbody` `\measuretbody{ $\langle body \rangle$ }` measures the tabular  $\langle body \rangle$ .

```

5559 \newcommand{\measuretbody}[1]{%
5560   \disable@notes%
5561   \disablel@dtabfeet%
5562   \l@dcolcount=0%
5563   \nullsetzen%
5564   \l@dcolcount=0
5565   \measuretbody #1\\&\\%
5566   \restore@notes%
5567   \global\l@dampcount=1}
5568
```

`\ltab` Array with entries left justified.

```

\edbeforetab 5569 \newcommand{\ltab}[1]{%
\edaftertab 5570   \l@dnnullfills
5571   \def\edbeforetab##1##2{\leftltab{##1}{##2}}%
```

```

5572 \def\edaftertab##1##2{\rightltab{##1}{##2}}%
5573 \measuretbody{#1}%
5574 \l@drestorefills
5575 \variab
5576 \setmrowleft #1\\&\\%
5577 \enablel@dtabfeet}
5578

```

`\ltabtext` Tabular with entries left justified.

```

5579 \newcommand{\ltabtext}[1]{%
5580 \l@dnnullfills
5581 \measuretbody{#1}%
5582 \l@drestorefills
5583 \variab
5584 \settrrowleft #1\\&\\%
5585 \enablel@dtabfeet}
5586

```

`\ctab` Array with centered entries.

```

\edbeforetab 5587 \newcommand{\ctab}[1]{%
\edaftertab 5588 \l@dnnullfills
5589 \def\edbeforetab##1##2{\leftctab{##1}{##2}}%
5590 \def\edaftertab##1##2{\rightctab{##1}{##2}}%
5591 \measuretbody{#1}%
5592 \l@drestorefills
5593 \variab
5594 \setmrowcenter #1\\&\\%
5595 \enablel@dtabfeet}
5596

```

`\ctabtext` Tabular with entries centered.

```

5597 \newcommand{\ctabtext}[1]{%
5598 \l@dnnullfills
5599 \measuretbody{#1}%
5600 \l@drestorefills
5601 \variab
5602 \settrrowcenter #1\\&\\%
5603 \enablel@dtabfeet}
5604

```

`\spreadtext`

```

5605 \newcommand{\spreadtext}[1]{%\l@dcolcount=\l@dampcount%
5606 \hb@xt@ \the\l@dcolwidth{\hbox{#1}\hss}}

```

`\spreadmath`

```

5607 \newcommand{\spreadmath}[1]{%
5608 \hb@xt@ \the\l@dcolwidth{\hbox{$\displaystyle{#1}$}\hss}}
5609

```



\HILFSskip More helpers.

\Hilfsskip 5610 \newskip\HILFSskip  
5611 \newskip\Hilfsskip  
5612

\EDTABINDENT

```
5613 \newcommand{\EDTABINDENT}{%
5614   \ifnum\l@dcolcount=30\let\NEXT\relax\l@dcolcount=0%
5615   \else\step1@dcolcount%
5616     \advance\Hilfsskip by\l@dcolwidth%
5617     \ifdim\l@dcolwidth=0pt\advance\hilfscount\@ne
5618     \else\advance\Hilfsskip by \the\hilfscount\edtabcolsep%
5619     \hilfscount=1\fi%
5620     \let\NEXT=\EDTABINDENT%
5621   \fi\NEXT}%
```

\edtabindent (was \tabindent)

```
5622 \newcommand{\edtabindent}{%
5623   \l@dcolcount=0\relax
5624   \Hilfsskip=0pt%
5625   \hilfscount=1\relax
5626   \EDTABINDENT%
5627   \hilfsskip=\hsize%
5628   \advance\hilfsskip -\Hilfsskip%
5629   \Hilfsskip=0.5\hilfsskip%
5630 }%
5631
```

\EDTAB (was \TAB)

```
5632 \def\EDTAB #1|#2|{%
5633   \setbox\tabhilfbox=\hbox{$\displaystyle{#1}$}%
5634   \setbox\tabHilfbox=\hbox{$\displaystyle{#2}$}%
5635   \advance\tabelskip -\wd\tabhilfbox%
5636   \advance\tabelskip -\wd\tabHilfbox%
5637   \unhbox\tabhilfbox\hskip\tabelskip%
5638   \unhbox\tabHilfbox}%
5639
```

\EDTABtext (was \TABtext)

```
5640 \def\EDTABtext #1|#2|{%
5641   \setbox\tabhilfbox=\hbox{#1}%
5642   \setbox\tabHilfbox=\hbox{#2}%
5643   \advance\tabelskip -\wd\tabhilfbox%
5644   \advance\tabelskip -\wd\tabHilfbox%
5645   \unhbox\tabhilfbox\hskip\tabelskip%
5646   \unhbox\tabHilfbox}%
```

\tabhilfbox Further helpers.

\tabHilfbox 5647 \newbox\tabhilfbox

```
5648 \newbox\tabHilfbox
5649
```

## XXVIII.2.4 Environments

```
edarrayl  The 'environment' forms for \ltab, \ctab and \rtab.
edarrayc  5650 \newenvironment{edarrayl}{\l@dcollect@body\ltab}{\l@}
edarrayr  5651 \newenvironment{edarrayc}{\l@dcollect@body\ctab}{\l@}
          5652 \newenvironment{edarrayr}{\l@dcollect@body\rtab}{\l@}
          5653

edtabularl  The 'environment' forms for \ltabtext, \ctabtext and \rtabtext.
edtabularc  5654 \newenvironment{edtabularl}{\l@dcollect@body\ltabtext}{\l@}
edtabularr  5655 \newenvironment{edtabularc}{\l@dcollect@body\ctabtext}{\l@}
          5656 \newenvironment{edtabularr}{\l@dcollect@body\rtabtext}{\l@}
          5657
```

## XXIX Quotation's commands

`\initnumbering@quote` This macro, called at the begin of any numbered section, redefine locally the quotation and quote environment, in order to allow to use it inside numbered section.

```
\quotation  \initnumbering@quote defines quotation environment.
\endquotation  5658 \newcommand{\initnumbering@quote}{\l@}
\quote  5659 \ifnoquotation@{\else
\endquote  5660 \renewcommand{\quotation}{\par\leavevmode%
          5661 \parindent=1.5em%
          5662 \skipnumbering%
          5663 \ifautopar%
          5664 \vskip-\parskip%
          5665 \else%
          5666 \vskip\topsep%
          5667 \fi%
          5668 \global\leftskip=\leftmargin%
          5669 \global\rightskip=\leftmargin%
          5670 }
          5671 \renewcommand{\endquotation}{\par%
          5672 \global\leftskip=0pt%
          5673 \global\rightskip=0pt%
          5674 \leavevmode%
          5675 \skipnumbering%
          5676 \ifautopar%
          5677 \vskip-\parskip%
          5678 \else%
          5679 \vskip\topsep%
          5680 \fi%
          5681 }
          5682 \renewcommand{\quote}{\par\leavevmode%
```

```

5683             \parindent=0pt%
5684             \skipnumbering%
5685             \ifautopar%
5686                 \vskip-\parskip%
5687             \else%
5688                 \vskip\topsep%
5689             \fi%
5690             \global\leftskip=\leftmargin%
5691             \global\rightskip=\leftmargin%
5692     }
5693     \renewcommand{\endquote}{\par%
5694             \global\leftskip=0pt%
5695             \global\rightskip=0pt%
5696             \leavevmode%
5697             \skipnumbering%
5698             \ifautopar%
5699                 \vskip-\parskip%
5700             \else%
5701                 \vskip\topsep%
5702             \fi%
5703     }
5704     \fi
5705 }

```

## XXX Section's title commands

### XXX.1 Commands to disable some feature

`\ledsectnotoc` The `\ledsectnotoc` only disables the `\addcontentsline` macro.

```
5706 \newcommand{\ledsectnotoc}{\let\addcontentsline\@gobblethree}
```

`\ledsectnomark` The `\ledsectnomark` only disables the `\chaptermark`, `\sectionmark` and `\subsectionmark` macros.

```

5707 \newcommand{\ledsectnomark}{%
5708     \let\chaptermark\@gobble%
5709     \let\sectionmark\@gobble%
5710     \let\subsectionmark\@gobble%
5711 }

```

### XXX.2 General overview

The system of `\eledxxxx` commands to section text work like this:

1. When one of these commands is called, `reledmac` 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. reledmac 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.

### XXX.3 `\beforeeledchapter` command

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`. As we patch command inside this test, we need to change the category code of # character *before* `\notbool` statement, because the second argument is read with the standard `catcode` (read *The TeXbook* to understand when the `catcode`'s change has effect).

```
5712 \catcode`\#=12
5713 \notbool{@noeled@sec}{%
```

`\beforeeledchapter` For technical reasons, not yet solved, page-breaking before chapters can't be made automatically by `eledmac`. Users have to use `\beforeeledchapter`.

```
5714 \ifl@dmemoir
5715   \newcommand\beforeeledchapter{%
5716     \clearforchapter%
5717   }
5718 \else
5719   \newcommand\beforeeledchapter{%
5720     \if@openright%
5721       \cleardoublepage%
5722     \else%
5723       \clearpage%
5724     \fi%
5725   }
5726 \fi
```

### XXX.4 Auxiliary commands

`\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.

```
5727 \newif\if@eled@sectioning
```

`\print@leftmargin@eledsection` and `\print@rightmargin@eledsection` are added by `reledmac` inside the code of sectioning command, in order to affix lines numbers. They include tests for RTL languages.

```

5728 \def\print@rightmargin@eledsection{%
5729   \if@eled@sectioning%
5730     \begingroup%
5731     \if@RTL%
5732       \let\llap\rlap%
5733       \let\leftlinenum\rightlinenum%
5734       \let\leftlinenumR\rightlinenumR%
5735       \let\l@drd@ta\l@dld@ta%
5736       \let\l@drsn@te\l@dlsn@te%
5737     \fi%
5738     \hfill\l@drd@ta \csuse{LR}{\l@drsn@te}%
5739   \endgroup%
5740 \fi%
5741 }%
5742
5743 \def\print@leftmargin@eledsection{%
5744   \if@eled@sectioning%
5745     \leavevmode%
5746     \begingroup%
5747     \if@RTL%
5748       \let\rlap\llap%
5749       \let\rightlinenum\leftlinenum%
5750       \let\rightlinenumR\leftlinenumR%
5751       \let\l@dld@ta\l@drd@ta%
5752       \let\l@dlsn@te\l@drsn@te%
5753     \fi%
5754     \l@dld@ta\csuse{LR}{\l@dlsn@te}%
5755   \endgroup%
5756 \fi%
5757 }%
5758

```

### XXX.5 Patching standard commands

`\chapter` We have to patch  $\LaTeX$ , book and memoir sectioning commands in order to:

- `\M@sect` • Disable `\edtext` inside.
- `\@mem@old@ssect` • Disable page breaking (for `\chapter`).
- `\@makechapterhead` • Add line numbers and sidenotes.
- `\@makechapterhead`
- `\@makeschapterhead`

`\@sect` Unfortunately, Maïeul Rouquette was not able to try if memoir is loaded. That is why  
`\@ssect` `eledmac` tries to define for both standard class and memoir class.

```

5759 \AtBeginDocument{%
5760 \patchcmd{\chapter}{\clearforchapter}{%
5761   \if@eled@sectioning\else%

```

```

5762     \ifl@dprintingpages\else%
5763         \clearforchapter%
5764     \fi%
5765 \fi%
5766 }
5767 {}
5768 {}
5769
5770 \pretocmd{\M@sect}
5771     {\let\old@edtext=\edtext%
5772     \let\edtext=\dummy@edtext@showlemma%
5773     }
5774 {}
5775 {}
5776
5777 \apptocmd{\M@sect}
5778     {\let\edtext=\old@edtext}
5779 {}
5780 {}
5781
5782 \patchcmd{\M@sect}
5783     { #9}
5784     { #9%
5785     \print@rightmargin@eledsection%
5786     }
5787 {}
5788 {}
5789
5790 \patchcmd{\M@sect}
5791     {\hskip #3\relax}
5792     {\hskip #3\relax%
5793     \print@leftmargin@eledsection%
5794     }
5795 {}
5796 {}
5797
5798 \patchcmd{\@mem@old@ssect}
5799     {#5}
5800     {#5%
5801     \print@leftmargin@eledsection%
5802     }
5803 {}
5804 {}
5805
5806 \patchcmd{\@mem@old@ssect}
5807     {\hskip #1}
5808     {\hskip #1%
5809     \print@rightmargin@eledsection%
5810     }
5811 {}

```

```

5812 {}
5813
5814 \patchcmd{\chapter}{\if@openright\cleardoublepage\else\clearpage\fi}{%
5815   \if@eled@sectioning\else%
5816     \ifl@dprintingpages\else%
5817       \if@openright\cleardoublepage\else\clearpage\fi%No clearpage inside a \eledsection: will keep criti
5818     \fi%
5819   \fi%
5820 }%
5821 {}%
5822 {}%
5823
5824 \patchcmd{\@makechapterhead}
5825   {#1}
5826   {\print@leftmargin@eledsection%
5827     #1%
5828     \print@rightmargin@eledsection%
5829   }
5830 {}
5831 {}
5832
5833 \patchcmd{\@makechapterhead}% For BIDI
5834   {\if@RTL\raggedleft\else\raggedright\fi}%
5835   {\if@eled@sectioning\else%
5836     \if@RTL\raggedleft\else\raggedright\fi%
5837   \fi%
5838   }%
5839   {}%
5840   {}%
5841
5842 \patchcmd{\@makeschapterhead}
5843   {#1}
5844   {\print@leftmargin@eledsection%
5845     #1%
5846     \print@rightmargin@eledsection%
5847   }
5848 {}
5849 {}
5850
5851 \pretocmd{\@sect}
5852   {\let\old@edtext=\edtext
5853     \let\edtext=\dummy@edtext@showlemma%
5854   }
5855 {}
5856 {}
5857
5858 \apptocmd{\@sect}
5859   {\let\edtext=\old@edtext}
5860 {}
5861 {}

```

```

5862
5863 \pretocmd{\@ssect}
5864   {\let\old@edtext=\edtext%
5865    \let\edtext=\dummy@edtext@showlemma%
5866   }
5867   {}
5868   {}
5869
5870 \apptocmd{\@ssect}
5871   {\let\edtext=\old@edtext}
5872   {}
5873   {}
5874

```

hyperref also redefines \@sect. That is why, when manipulating arguments, we patch \@sect and the same only if hyperref is not used. If it is, we patch the \NR commands.

```

5875 \@ifpackageloaded{nameref}{
5876
5877   \patchcmd{\NR@sect}
5878     {#8}
5879     {#8%
5880      \print@rightmargin@eledsection%
5881     }
5882     {}
5883     {}
5884
5885   \patchcmd{\NR@sect}
5886     {\hskip #3\relax}
5887     {\hskip #3\relax%
5888      \print@leftmargin@eledsection%
5889     }
5890     {}
5891     {}
5892
5893   \patchcmd{\NR@ssect}
5894     {#5}
5895     {#5%
5896      \print@rightmargin@eledsection%
5897     }
5898     {}
5899     {}
5900
5901   \patchcmd{\NR@ssect}
5902     {\hskip #1}
5903     {\hskip #1%
5904      \print@leftmargin@eledsection%
5905     }
5906     {}
5907     {}
5908   }%

```



```

5909 {
5910 \patchcmd{\@sect}
5911   {#8}
5912   {#8%
5913   \print@rightmargin@eledsection%
5914   }
5915   {}
5916   {}
5917
5918 \patchcmd{\@sect}
5919   {\hskip #3\relax}
5920   {\hskip #3\relax%
5921   \print@leftmargin@eledsection%
5922   }
5923   {}
5924   {}
5925
5926 \patchcmd{\@ssect}
5927   {#5}
5928   {#5%
5929   \print@rightmargin@eledsection%
5930   }
5931   {}
5932   {}
5933
5934 \patchcmd{\@ssect}
5935   {\hskip #1}
5936   {\hskip #1%
5937   \print@leftmargin@eledsection%
5938   }
5939   {}
5940   {}
5941 }%
5942 }

```

Now, we have finished to patch the commands, using # with a catcode equals to 12. We close the \notbool{@noeled@sec} statement, restore the normal catcode for # and reopen a new \notbool{@noeled@sec} statement.

```

5943 {}}%
5944 \protect\catcode`\#=6 %Space NEEDS by \catcode
5945 \notbool{@noeled@sec}{%

```

### XXX.6 Main code of \eledxxx commands

\eled@sectioning@out \eled@sectioning@out is the output file, to dump the pstarts where a sectioning command is used.

```

5946 \newwrite\eled@sectioning@out

```

```

\eledchapter And now, the user sectioning commands, which write to the file, and also add content
\eledsection as a "normal" line.
\eledsubsection
\eledsubsubsection
\eledchapter*
\eledsection*
\eledsubsection*
\eledsubsubsection*

```

```

5947 \newcommand{\eledchapter}[2] [] {%
5948   #2%
5949   \ifledRcol%
5950     \immediate\write\eled@sectioningR@out{%
5951       \string\eled@chapter{#1}{\unexpanded{#2}}{\the\l@dumpstartsR}{R}
5952     }%
5953   \else%
5954     \immediate\write\eled@sectioning@out{%
5955       \string\eled@chapter{#1}{\unexpanded{#2}}{\the\l@dumpstartsL}{L}
5956     }%
5957   \fi%
5958 }
5959
5960 \newcommand{\eledsection}[2] [] {%
5961   #2%
5962   \ifledRcol%
5963     \immediate\write\eled@sectioningR@out{%
5964       \string\eled@section{#1}{\unexpanded{#2}}{\the\l@dumpstartsR}{R}
5965     }%
5966   \else%
5967     \immediate\write\eled@sectioning@out{%
5968       \string\eled@section{#1}{\unexpanded{#2}}{\the\l@dumpstartsL}{L}
5969     }%
5970   \fi%
5971 }
5972
5973 \newcommand{\eledsubsection}[2] [] {%
5974   #2%
5975   \ifledRcol%
5976     \immediate\write\eled@sectioningR@out{%
5977       \string\eled@subsection{#1}{\unexpanded{#2}}{\the\l@dumpstartsR}{R}
5978     }%
5979   \else%
5980     \immediate\write\eled@sectioning@out{%
5981       \string\eled@subsection{#1}{\unexpanded{#2}}{\the\l@dumpstartsL}{L}
5982     }%
5983   \fi%
5984 }
5985 \newcommand{\eledsubsubsection}[2] [] {%
5986   #2%
5987   \ifledRcol%
5988     \immediate\write\eled@sectioningR@out{%
5989       \string\eled@subsubsection{#1}{\unexpanded{#2}}{\the\l@dumpstartsR}{R}
5990     }%
5991   \else%
5992     \immediate\write\eled@sectioning@out{%
5993       \string\eled@subsubsection{#1}{\unexpanded{#2}}{\the\l@dumpstartsL}{L}
5994     }%
5995   \fi%
5996 }

```

```

5997
5998
5999 \WithSuffix\newcommand\eledchapter*[2] [] {%
6000   #2%
6001   \ifledRcol%
6002     \immediate\write\eled@sectioningR@out{%
6003       \string\eled@chapter{#1}{\unexpanded{#2}}{\the\l@dumpstartsR}{*}{R}
6004     }%
6005   \else%
6006     \immediate\write\eled@sectioning@out{%
6007       \string\eled@chapter{#1}{\unexpanded{#2}}{\the\l@dumpstartsL}{*}{L}
6008     }%
6009   \fi%
6010 }
6011
6012 \WithSuffix\newcommand\eledsection*[2] [] {%
6013   #2%
6014   \ifledRcol%
6015     \immediate\write\eled@sectioningR@out{%
6016       \string\eled@section{#1}{\unexpanded{#2}}{\the\l@dumpstartsR}{*}{R}
6017     }%
6018   \else%
6019     \immediate\write\eled@sectioning@out{%
6020       \string\eled@section{#1}{\unexpanded{#2}}{\the\l@dumpstartsL}{*}{L}
6021     }%
6022   \fi%
6023 }
6024
6025 \WithSuffix\newcommand\eledsubsection*[2] [] {%
6026   #2%
6027   \ifledRcol%
6028     \immediate\write\eled@sectioningR@out{%
6029       \string\eled@subsection{#1}{\unexpanded{#2}}{\the\l@dumpstartsR}{*}{R}
6030     }%
6031   \else%
6032     \immediate\write\eled@sectioning@out{%
6033       \string\eled@subsection{#1}{\unexpanded{#2}}{\the\l@dumpstartsL}{*}{L}
6034     }%
6035   \fi%
6036 }
6037
6038 \WithSuffix\newcommand\eledsubsubsection*[2] [] {%
6039   #2%
6040   \ifledRcol%
6041     \immediate\write\eled@sectioningR@out{%
6042       \string\eled@subsubsection{#1}{\unexpanded{#2}}{\the\l@dumpstartsR}{*}{R}
6043     }%
6044   \else%
6045     \immediate\write\eled@sectioning@out{%
6046       \string\eled@subsubsection{#1}{\unexpanded{#2}}{\the\l@dumpstartsL}{*}{L}

```

```

6047     }%
6048   \fi%
6049 }

```

### XXX.7 Macros written in the auxiliary file

<pre> \eled@chapter \eled@section \eled@subsection \eled@subsubsection </pre>	<p>The sectioning macros, called in the auxiliary file. They have five arguments:</p> <ol style="list-style-type: none"> <li>1. Optional arguments of <math>\LaTeX</math> sectioning command.</li> <li>2. Mandatory arguments of <math>\LaTeX</math> sectioning command.</li> <li>3. Pstart number.</li> <li>4. Side: R if right, nothing if left.</li> <li>5. Starred or not.</li> </ol>
---	---

```

6050 \def\eled@chapter#1#2#3#4#5{%
6051   \ifstrempy{#4}%
6052   {%
6053     \ifstrempy{#1}%
6054     {%
6055       \global\csdef{eled@sectioning@#3#5}{\let\edtext=\dummy@edtext@showlemma\chapter{#2}}%
6056       \global\csdef{eled@sectmark@#3#5}{\let\edtext=\dummy@edtext{}\chaptermark{#2}}%
6057       }%Need for \pairs, because of using parbox.
6058     {%
6059       \global\csdef{eled@sectioning@#3#5}{\let\edtext=\dummy@edtext@showlemma\chapter[#1]{#2}}%
6060       \global\csdef{eled@sectmark@#3#5}{\let\edtext=\dummy@edtext{}\chaptermark{#2}}%Need 1
6061     }%
6062   }%
6063   {%
6064     \ifstrempy{#1}%
6065     {\global\csdef{eled@sectioning@#3#5}{\let\edtext=\dummy@edtext@showlemma\chapter*{#2}}%
6066     {\global\csdef{eled@sectioning@#3#5}{\let\edtext=\dummy@edtext@showlemma\chapter*{#1}}%
6067     }%
6068     \listcsadd{eled@sections#5@@}{#3}%
6069   }
6070 \def\eled@section#1#2#3#4#5{%
6071   \ifstrempy{#4}%
6072   {\ifstrempy{#1}%
6073     {%
6074       \global\csdef{eled@sectioning@#3#5}{\section{#2}}%
6075       \global\csdef{eled@sectmark@#3#5}{\let\edtext=\dummy@edtext{}\sectionmark{#2}}%Need 1
6076     }%
6077     {%
6078       \global\csdef{eled@sectioning@#3#5}{\section[#1]{#2}}%
6079       \global\csdef{eled@sectmark@#3#5}{\let\edtext=\dummy@edtext{}\sectionmark{#1}}%Need 1
6080     }%
6081   }%
6082   {\ifstrempy{#1}%

```

```

6083     {\global\csdef{eled@sectioning@#3#5}{\section*{#2}}}%
6084     {\global\csdef{eled@sectioning@#3#5}{\section*{#1}{#2}}}%Bug in LaTeX!
6085     }
6086 \listcsadd{eled@sections#5@@}{#3}%
6087 }
6088 \def\eled@subsection#1#2#3#4#5{%
6089     \ifstrempy{#4}%
6090     {\ifstrempy{#1}%
6091     {%
6092         \global\csdef{eled@sectioning@#3#5}{\subsection{#2}}%
6093         \global\csdef{eled@sectmark@#3#5}{\let\edtext=\dummy@edtext{}\csuse{subsectionmark}{#2}}}%Need fo
6094     }%
6095     {%
6096         \global\csdef{eled@sectioning@#3#5}{\subsection[#1]{#2}}%
6097         \global\csdef{eled@sectmark@#3#5}{\let\edtext=\dummy@edtext{}\csuse{subsectionmark}{#1}}}%Need fo
6098     }%
6099     }%
6100     {\ifstrempy{#1}%
6101     {\global\csdef{eled@sectioning@#3#5}{\subsection*{#2}}}%
6102     {\global\csdef{eled@sectioning@#3#5}{\subsection*{#1}{#2}}}%Bug in LaTeX!
6103     }
6104 \listcsadd{eled@sections#5@@}{#3}%
6105 }
6106 \def\eled@subsubsection#1#2#3#4#5{%
6107     \ifstrempy{#4}%
6108     {\ifstrempy{#1}%
6109     {\global\csdef{eled@sectioning@#3#5}{\subsubsection{#2}}}%
6110     {\global\csdef{eled@sectioning@#3#5}{\subsubsection[#1]{#2}}}%
6111     }%
6112     {\ifstrempy{#1}%
6113     {\global\csdef{eled@sectioning@#3#5}{\subsubsection*{#2}}}%
6114     {\global\csdef{eled@sectioning@#3#5}{\subsubsection*{#1}{#2}}}%Bug in LaTeX!
6115     }
6116 \listcsadd{eled@sections#5@@}{#3}%
6117 }
6118

```

End of the conditional test about noeledsec option.

```
6119 }{}
```

## XXXI 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.

`\normal@page@break` `\normal@page@break` is an etoolbox list which contains the absolute line number of the last line, for each page.

```
6120 \def\normal@page@break{}
```

`\prev@pb` The `\l@prev@pb` macro is a etoolbox list, which contains the lines in which page breaks occur (before or after). The `\l@prev@nopb` macro is a etoolbox list, which contains the lines with NO page break before or after.

```
6121 \def\l@prev@pb{}
6122 \def\l@prev@nopb{}

\ledpb The \ledpb macro writes the call to \led@pb in line-list file. The \ledpbnum macro
\ledpbnum writes the call to \led@pbnum in line-list file. The \lednopb macro writes the call to
\lednopb \led@nopb in line-list file. The \lednopbnum macro writes the call to \led@nopbnum
\lednopbnum in line-list file.
6123 \newcommand{\ledpb}{\write\linenum@out{\string\led@pb}}
6124 \newcommand{\ledpbnum}[1]{\write\linenum@out{\string\led@pbnum{#1}}}
6125 \newcommand{\lednopb}{\write\linenum@out{\string\led@nopb}}
6126 \newcommand{\lednopbnum}[1]{\write\linenum@out{\string\led@nopbnum{#1}}}
```

`\led@pb` The `\led@pb` adds the absolute line number in the `\prev@pb` list. The `\led@pbnum` adds the argument in the `\prev@pb` list. The `\led@nopb` adds the absolute line number in the `\prev@nopb` list. The `\led@nopbnum` adds the argument in the `\prev@nopb` list.

```
\led@pbnum 6127 \newcommand{\led@pb}{\listxadd{\l@prev@pb}{\the\absline@num}}
\led@pbnum 6128 \newcommand{\led@pbnum}[1]{\listxadd{\l@prev@pb}{#1}}
\led@nopb 6129 \newcommand{\led@nopb}{\listxadd{\l@prev@nopb}{\the\absline@num}}
\led@nopbnum 6130 \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.

```
\led@pb@setting 6131 \def\led@pb@setting{before}
6132 \newcommand{\ledpbsetting}[1]{\gdef\led@pb@setting{#1}}
```

`\led@check@pb` The `\led@check@pb` and `\led@check@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 `\l@pb`.

```
\led@check@nopb 6133 \newcommand{\led@check@pb}{\xifinlist{\the\absline@num}{\l@prev@pb}{\pagebreak[4]}}
6134 \newcommand{\led@check@nopb}{%
6135   \IfStrEq{\led@pb@setting}{before}{%
6136     \xifinlist{\the\absline@num}{\l@prev@nopb}%
6137     {\numdef{\abs@prevline}{\the\absline@num-1}%
6138     \xifinlist{\abs@prevline}{\normal@page@break}%
6139     {\nopagebreak[4]\enlargethispage{\baselineskip}}%
6140     }%
6141   }%
6142   }%
6143   }%
6144   \IfStrEq{\led@pb@setting}{after}{%
6145     \xifinlist{\the\absline@num}{\l@prev@nopb}{%
```

```

6146      \xifinlist{\the\absline@num}{\normal@page@break}%
6147      {\nopagebreak[4]\enlargethispage{\baselineskip}}%
6148      {}%
6149 }%
6150     {}%
6151     {}%
6152     {}%
6153 }

```

## XXXII Long verse: prevents being separated by a page break

`\iflednopbinverse` The `\lednopbinverse` boolean is set to false by default. If set to true, `reledmac` 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.

`\check@pb@in@verse` The `\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 `\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 `\led@nopb` list, if the page break must occur after the verse.

```

6154 \newcommand{\check@pb@in@verse}{%
6155   \ifinstanza\iflednopbinverse\ifinserthangingsymbol% Using stanzas and enabling page breaks in verse c
6156   \ifnum\page@num=\last@page@num\else%If we have change page
6157     \IfStrEq{\led@pb@setting}{before}{%
6158       \numgdef{\abs@line@verse}{\the\absline@num-1}%
6159       \ledpbnum{\abs@line@verse}%
6160     }{}%
6161     \IfStrEq{\led@pb@setting}{after}{%
6162       \numgdef{\abs@line@verse}{\the\absline@num-1}%
6163       \lednopbnum{\abs@line@verse}%
6164     }{}%
6165   \fi%
6166   \fi\fi\fi%
6167 }

```

## XXXIII Compatibility with `eledmac`

Here, we define some command for the `eledmac-compat` option.

```

6168 \ifeledmaccompat%
6169
6170 \newcommand{\footnormalX}[1]{\arrangementX[#1]{normal}}%
6171 \newcommand{\footparagraphX}[1]{\arrangementX[#1]{paragraph}}%
6172 \newcommand{\foottwocolX}[1]{\arrangementX[#1]{twocol}}%

```

```

6173 \newcommand{\footthreecolX}[1]{\XarrangementX[#1]{threecol}}%
6174
6175 \unless\ifnocritical@
6176 \newcommand{\footnormal}[1]{\Xarrangement[#1]{normal}}%
6177 \newcommand{\footparagraph}[1]{\Xarrangement[#1]{paragraph}}%
6178 \newcommand{\foottwocol}[1]{\Xarrangement[#1]{twocol}}%
6179 \newcommand{\footthreecol}[1]{\Xarrangement[#1]{threecol}}%
6180 \let\hsizetwocol\Xhsizetwocol
6181 \let\hsizethreecol\Xhsizethreecol
6182 \let\bhookXnote\Xbhooknote
6183 \let\boxsymlinenum\Xboxsymlinenum
6184 \let\symlinenum\Xsymlinenum
6185 \let\beforenumberinfootnote\Xbeforenumber
6186 \let\afternumberinfootnote\Xafternumber
6187 \let\beforeXsymlinenum\XbeforeXsymlinenum
6188 \let\afterXsymlinenum\XafterXsymlinenum
6189 \let\inplaceofnumber\Xinplaceofnumber
6190 \let\Xlemmaseparator\lemmaseparator
6191 \let\afterlemmaseparator\Xafterlemmaseparator
6192 \let\beforelemmaseparator\Xbeforelemmaseparator
6193 \let\inplaceoflemmaseparator\Xinplaceoflemmaseparator
6194 \let\txbeforeXnotes\Xtxbeforenotes
6195 \let\afterXrule\Xafterrule
6196 \let\numberonlyfirstinline\Xnumberonlyfirstinline
6197 \let\numberonlyfirstintwoline\Xnumberonlyfirstintwoline
6198 \let\nonnumberinfootnote\Xnonnumberinfootnote
6199 \let\pstartinfootnote\Xpstart
6200 \let\pstartinfootnoteeverytime\Xpstarteverytime
6201 \let\onlyXpstart\Xonlypstart
6202 \let\Xnonnumberinfootnote\Xnonnumber
6203 \let\nonbreakableafternumber\Xnonbreakableafternumber
6204 \let\maxhXnotes\Xmaxhnotes
6205 \let\beforeXnotes\Xbeforenotes
6206 \let\boxlinenum\Xboxlinenum
6207 \let\boxlinenumalign\Xboxlinenumalign
6208 \let\boxstartlinenum\Xboxstartlinenum
6209 \let\boxendlinenum\Xboxendlinenum
6210 \let\twoline\Xtwoline
6211 \let\morethantwoline\Xmorethantwoline
6212 \let\twolinebutnotmore\Xtwolinebutnotmore
6213 \let\twolineonlyinsamepage\Xtwolineonlyinsamepage
6214 \fi
6215
6216 \unless\ifnofamiliar@
6217 \let\notesXwidthliketwocolumns\noteswidthliketwocolumnsX
6218 \fi
6219 \newcommandx[2][1,usedefault]{\parafootsep}{%
6220 \Xparafootsep[#1]{#2}%
6221 \parafootsepX[#1]{#2}
6222 }%

```



```

6223
6224 \newcommandx[2][1,usedefault]{\afternote}{%
6225     \Xafternote[#1]{#2}%
6226     \afternoteX[#1]{#2}%
6227 }%
6228
6229 \unless\ifnoend@
6230 \let\XendXtwolines\Xendtwolines
6231 \let\XendXmorethantwolines\Xendmorethantwolines
6232 \let\XhookXendnote\Xendbhooknote
6233 \let\boxXendlinenum\Xendboxlinenum%
6234 \let\boxXendlinenumalign\Xendboxlinenumalign%
6235 \let\boxXendstartlinenum\Xendboxstartlinenum%
6236 \let\boxXendendlinenum\Xendboxendlinenum%
6237 \let\XendXlemmaseparator\Xendlemmaseparator
6238 \let\XendXbeforelemmaseparator\Xendbeforelemmaseparator
6239 \let\XendXafterlemmaseparator\Xendafterlemmaseparator
6240 \let\XendXinplaceoflemmaseparator\Xendinplaceoflemmaseparator
6241 \fi
6242
6243 \AtBeginDocument{%
6244     \ifdef\lineref{}\let\lineref\edlineref}%
6245 }%
6246
6247
6248 \fi%
</code>

```

## Appendix A Some things to do when changing version

### Appendix A.1 Migrating from edmac to ledmac

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 ledmac.

The package still provides the original `\text` command, but it is (a) deprecated, and (b) its name has been changed<sup>33</sup> to `\critext`; use the `\edtext` macro instead. However, if you do use `\critext` (the new name for `\text`), the following is a reminder.

`\critext` Within numbered paragraphs, footnotes and endnotes are generated by forms of the `\critext` macro:

```
\critext{⟨lemma⟩}⟨commands⟩/
```

The `⟨lemma⟩` argument is the lemma in the main text: `\critext` both prints this as part of the text, and makes it available to the `⟨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:

I saw my friend <code>\critext{Smith}</code>	1 I saw my friend
<code>\Afootnote{Jones C, D.}/</code>	2 Smith on Tuesday.
on Tuesday.	<u>2 Smith</u> ] Jones C, D.

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 `\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:

<code>\critext{I saw my friend</code>	1 I saw my friend
<code>\critext{Smith}{\Afootnote{Jones</code>	2 Smith on Tuesday.
<code>C, D.}/ on Tuesday.}</code>	<u>2 Smith</u> ] Jones C, D.
<code>\Bfootnote{The date was</code>	<u>1-2</u> I saw my friend
<code>July 16, 1954.}</code>	Smith on Tuesday.] The
<code>/</code>	date was July 16, 1954.

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

The second argument of the `\critext` macro, `⟨commands⟩`, is the same as the second argument to the `\edtext` macro.

It is possible to define aliases for `\critext`, which can be easier to type. You can make a single character substitute for `\critext` by saying this:

<sup>33</sup>A name like `\text` is likely to be defined by other  $\text{\LaTeX}$  packages (it certainly is by the AMS packages) and it seems sensible to try and avoid clashes with other definitions.

```
\catcode`\<=\active
\let<=\critext
```

Then you might say `<{Smith}\variant{Jones}/`. This of course destroys the ability to use `<` 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:

```
\catcode`\<=\active
\def\xtext#1#2>{\critext{#1}{#2}/}
\let<=\xtext
```

This allows you to say `<{Smith}\Afootnote{Jones}>`.

Aliases for `\critext` 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 `\critext`. (See VI p. 96 to find out why.) Aliases of the second kind may be nested without any problem.

If you really have to use `\critext` in any of the tabular or array environments, then `\edtext` must not be used in the same environment. If you use `\critext` in one of these environments then you have to issue the declaration `\usingcritext` beforehand. The declaration `\usingedtext` must be issued to revert to the default assumption that `\edtext` will be used.

## Appendix A.2 Migration from ledmac to eledmac

In eledmac, some changes were made in the code to allow easy customization. This may cause problems for people who have already made their own. The next sections explain how to handle this.

If you have created your own series using `\addfootins` and `\addfootinsX`, you must use instead the `\newseries` command (see 5.5.1 p. 28), and remove any `\Xfootnote` command.

If you have customized the `\XXXXXfmt` command, please check whether you can achieve the same by the commands documented for display options (6 p. 28) or `\Xfootnote` options (5.2.2 p. 22). Otherwise please add a new ticket on Github to request a new function for doing this.<sup>34</sup>

If for some reason you do not want to make the modifications to use the new functions of eledmac, you can continue using your own `\XXXXXfmt` command, but you must replace:

```
\renewcommand*{XXXXfmt}[3]
```

with

```
\renewcommandx*{XXXXfmt}[4][4=Z]
```

---

<sup>34</sup><https://github.com/maieul/ledmac/issues>

If you do not make that, you will get a spurious [X], where X is series letter.

If you used a `\protect` command inside a `\footnote` command inside a numbered section, you must change the `\protect` to `\noexpand`. Otherwise the command after the `\protect` will be discarded.

### Appendix A.3 Migration to eledmac 1.5.1

The version 1.5.1 corrects a bug in `stanzaindentsrepetition` (cf. 8.3 p. 39). This bug had two consequences:

1. `stanzaindentsrepetition` did not 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 a value equal to 2, you had to change your `\setstanzaindents`. Explanation:

```
\setcounter{stanzaindentsrepetition}{2}
\setstanzaindents{5,1,0}
```

This code, in versions prior to 1.5.1, made the first line have an indentation of 0, the second line of 1, the third verse of 0, the fourth verse of 1 and so forth.

But this code should have instead achieved quite the contrary: the first line would have an indentation of 1, the second line of 0, the third line of 1, the fourth line of 0 and so forth.

So version 1.5.1 corrected this bug. If you want to keep the former presentation, you must change:

```
\setcounter{stanzaindentsrepetition}{2}
\setstanzaindents{5,1,0}
```

to:

```
\setcounter{stanzaindentsrepetition}{2}
\setstanzaindents{5,0,1}
```

### Appendix A.4 Migration to eledmac 1.12.0

The migration to eledmac 1.12.0 is easy:

- You must first delete all the auxiliary files, then compile your document three times as usual.
- If you have modified `\l@reg`, which is not advisable, you must rename it to `\@nl@reg`.

There is an additional problem. If you have put text into brackets just after `\pstart` or `\pend`, this text will be considered to be an optional argument of `\pstart` or `\pend` (see 4.2.3 p. 16). If so, add a `\relax` between `\pstart`/`\pend` and the first bracket.

The version 1.12.0 also introduce a better way to handle sectional divisions inside numbered text. Please read 14.2 p. 52.

## Appendix A.5 Migration to eledmac 17.1

This version changes the default setting of `\Xpstart`. Henceforth, `pstart` numbers will be printed in footnotes within the section of text where you have called `\numberpstarttrue`.

We do not see any reason to print them in the other sections. However, if you want to print the `pstart` numbers in all of the footnotes, whatever the section, without having to use `\numberpstarttrue`, you can use `\Xpstarteverytime`.

## Appendix A.6 Migration to eledmac 1.21.0

### Appendix A.6.1 `\Xledsetnormalparstuff` and `\ledsetnormalparstuffX`

The `\ledsetnormalparstuff` has been split into two different commands:

- `\Xledsetnormalparstuff` for critical notes;
- `\ledsetnormalparstuffX` for familiar notes.

Both commands can take an optional argument which is the series letter. If you have redefined `\ledsetnormalparstuff` or any of the commands which call them, you must change them accordingly.

### Appendix A.6.2 Endnotes

In any case, delete the `.end` file before the next run.

The previous version of Eledmac had a bug: there were two spaces between the starting page number and the starting line number, but only one space between the ending page number and the ending line number.

As a matter of fact, 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, you may load the package with the `oldprintnpnumspace` option.

If you have redefined `\endprint`, you must:

- Contact us and 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 `\ledinnote` command now takes a first optional argument, which is the label for the hyperreference. If you have redefined it, change your redefinition, and check whether you can avoid this redefinition by only redefining `\ledinnotemark`.

## Appendix A.8 Migration to eledmac 1.23.0

You must delete the numbered auxiliary files before compiling with the new version of eledmac.

## Appendix A.9 Migration from eledmac to reledmac

There are many changes in reledmac which require the user to make modifications.

### Appendix A.9.1 Risk of ‘no room for a new’

The risk to obtain a ‘no room for a new something’ error is greater in reledmac than it is in eledmac. See 17.2 p. 54 in order to know how to limit it.

### Appendix A.9.2 Multiple indices with memoir

Eledmac and ledmac used the specific indexing tools of the memoir class designed to produce multiple indices. However, eledmac could also use imakeidx or indextools tools independently of the memoir class. This system forced to maintain redundant code. Since reledmac, we use only the imakeidx or indextools tools.

Consequently: Users of memoir are invited to use indextool or imakeidx to produce multiple indices.

### Appendix A.9.3 Deprecated commands and options

The table of deprecated commands and their alternatives follows. Note that the way some commands must be used may have changed. Please read the handbook.

<i>Deprecated command</i>	<i>Replaced with</i>
<code>\addfootins</code>	<code>\newseries</code>
<code>\addfootinsX</code>	<code>\newseries</code>
<code>\critext</code>	<code>\edtext</code>
<code>\falseverse</code>	<code>\newverse</code>
<code>\interparanoteglue</code>	<code>\Xafternote</code> and <code>\afternoteX</code>
<code>\ledchapter</code>	<code>\eledchapter</code>
<code>\ledsection</code>	<code>\eledsection</code>
<code>\ledsetnormalparstuff</code>	<code>\Xledsetnormalparstuff</code> and <code>\ledsetnormalparstuffX</code>
<code>\ledsubsection</code>	<code>\eledsubsection</code>
<code>\ledsubsubsection</code>	<code>\eledsubsubsection</code>
<code>\noeledsec</code>	Package option <code>noeledsec</code>
<code>\noendnotes</code>	Package option <code>noendnotes</code>
<code>\pageparbreak</code>	<code>\ledpb</code>

The `ledsecnolinenumber` option has been removed, because it was related to deprecated commands.

The `oldprintnpnumspace` option has been removed too, because it was related to a historical bug. The `\usingedtext` and `\usingcritext` commands are also deprecated.

**Appendix A.9.4 \renewcommand replaced by command**

Many uses of \renewcommand have been replaced with uses of specific commands. Please read handbook about specific commands.

<i>Deprecated \renewcommand</i>	<i>Replaced with</i>
\@led@extranofeet	\newseries
\apprefprefixmore	\setapprefprefixmore
\apprefprefixsingle	\setapprefprefixsingle
\endstanzaextra	Optional argument of \&
\hangingsymbol	\sethangingsymbol
\ledfootinsdim	\Xmaxhnotes and \maxhnotesX
\parafootftmsep	\Xparafootsep and \parafootsepX
\notenumfont	\Xnotenumfont, \Xendnotenumfont and \notenumfontX
\notefontsetup	\Xnotefontsize, \Xendnotefontsize and \notefontsizeX
\sidenoteseq	\setsidenotsep
\startstanzahook	Optional argument of \stanza
\symplinenum	\Xsymplinenum

**Appendix A.9.5 Commands the names of which have been changed**

In order to help the migration from eledmac to reledmac, you may load reledmac with eledmac-compat option. However, it is advised not to, and to change the command names themselves instead. In many cases, you use only a few of them, except the \footparagraph command.

<i>Old command</i>	<i>New command</i>
\footparagraph	\Xarrangement
\footnormal	\Xarrangement
\foottwocol	\Xarrangement
\footthreecol	\Xarrangement
\footparagraphX	\arrangementX
\footnormalX	\arrangementX
\foottwocolX	\arrangementX
\footthreecolX	\arrangementX
\afterlemmaseparator	\Xafterlemmaseparator
\afternote	\Xafternote and \afternoteX
\afternumberinfootnote	\Xafternumber
\afterXrule	\Xafterrule
\afterXsymplinenum	\Xaftersymplinenum
\beforelemmaseparator	\Xbeforelemmaseparator
\beforenumberinfootnote	\Xbeforenumber
\beforeXnotes	\Xbeforenotes
\beforeXsymplinenum	\Xbeforesymplinenum

<i>Old command</i>	<i>New command</i>
<code>\bhookXnote</code>	<code>\Xbhookendnote</code>
<code>\bhookXnote</code>	<code>\Xbooknote</code>
<code>\boxendlinenum</code>	<code>\Xboxendlinenum</code>
<code>\boxlinenum</code>	<code>\Xboxlinenum</code>
<code>\boxlinenumalign</code>	<code>\Xboxlinenumalign</code>
<code>\boxstartlinenum</code>	<code>\Xboxstartlinenum</code>
<code>\boxsymlinenum</code>	<code>\Xboxsymlinenum</code>
<code>\boxXendlinenum</code>	<code>\Xendboxlinenum</code>
<code>\boxXendlinenumalign</code>	<code>\Xendboxlinenumalign</code>
<code>\boxXendstartlinenum</code>	<code>\boxXendstartlinenum</code>
<code>\letboxXendendlinenum</code>	<code>\Xendletboxendlinenum</code>
<code>\hsizetwocol</code>	<code>\Xhsizetwocol</code>
<code>\hsizethreecol</code>	<code>\Xhsizethreecol</code>
<code>\inplaceoflemmaseparator</code>	<code>\Xinplaceoflemmaseparator</code>
<code>\inplaceofnumber</code>	<code>\Xinplaceofnumber</code>
<code>\lemmaseparator</code>	<code>\Xlemmaseparator</code>
<code>\maxhXnotes</code>	<code>\Xmaxhnotes</code>
<code>\morethantwolines</code>	<code>\Xmorethantwolines</code>
<code>\nonumberinfootnote</code>	<code>\Xnonumber</code>
<code>\notesXwidthliketwocolumns</code>	<code>\noteswidthliketwocolumnsX</code>
<code>\noXlemmaseparator</code>	<code>\Xnolemmaseparator</code>
<code>\numberonlyfirstinline</code>	<code>\Xnumberonlyfirstinline</code>
<code>\numberonlyfirstintwolines</code>	<code>\Xnumberonlyfirstintwolines</code>
<code>\nonbreakableafternumber</code>	<code>\Xnonbreakableafternumber</code>
<code>\onlyXpstart</code>	<code>\Xonlypstart</code>
<code>\parafootsep</code>	<code>\Xparafootsep</code> and <code>\parafootsepX</code>
<code>\pstartinfootnote</code>	<code>\Xpstart</code>
<code>\pstartinfootnoteeverytime</code>	<code>\Xpstarteverytime</code>
<code>\symlinenum</code>	<code>\Xsymlinenum</code>
<code>\twolines</code>	<code>\Xtwolines</code>
<code>\twolinesbutnotmore</code>	<code>\Xtwolinesbutnotmore</code>
<code>\twolinesonlyinsamepage</code>	<code>\Xtwolinesonlyinsamepage</code>
<code>\txtbeforeXnotes</code>	<code>\Xtxtbeforenotes</code>
<code>\XendXafterlemmaseparator</code>	<code>\Xendafterlemmaseparator</code>
<code>\XendXbeforelemmaseparator</code>	<code>\Xendbeforelemmaseparator</code>
<code>\XendXinplaceoflemmaseparator</code>	<code>\Xendinplaceoflemmaseparator</code>
<code>\XendXlemmaseparator</code>	<code>\Xendlemmaseparator</code>
<code>\XendXmorethantwolines</code>	<code>\Xendmorethantwolines</code>
<code>\XendXtwolines</code>	<code>\Xendtwolines</code>
<code>\Xnonumberinfootnote</code>	<code>\Xnonumber</code>
<code>\lineref</code>	<code>\edlineref</code>



**Appendix A.9.6 Endnotes**

With *reledmac*, there is now one auxiliary file for every endnotes set (*.Aend*, *.Bend*, *.Cend* etc.). If you have overridden `\doendnotes` (which you would not have done) you must adapt your code.

**Appendix A.9.7 Z Series**

The ‘Z’ series of notes has been removed. Only five series are provided now by default: A, B, C, D, E.

**Appendix A.9.8 Internal commands**

Users who have overridden internal commands, which is wrong, must adapt according to the following. Or better, they should not override any of such commands and use *reledmac* options instead.

- If you have modified `\Xfootfmt`, note that the fourth argument is now mandatory.
- `\unvxh` has been replaced with `\Xunvxh` and `\unvxhX` with two mandatory arguments.

## References

- [Bre96] Herbert Breger. `tabmac`. October 1996. (Available from CTAN in `macros/plain/contrib/tabmac`)
- [Bur01] John Burt. ‘Typesetting critical editions of poetry’. *TUGboat*, **22**, 4, pp 353–361, December 2001. (Code available from CTAN in `macros/latex/contrib/poemscol`)
- [Eck03] Matthias Eckermann. *The Parallel-Package*. April 2003. (Available from CTAN in `macros/latex/contrib/parallel`)
- [Fai03] Robin Fairbairns. *footmisc — a portmanteau package for customising footnotes in L<sup>A</sup>T<sub>E</sub>X*. February 2003. (Available from CTAN in `macros/latex/contrib/footmisc`)
- [LW90] John Lavagnino and Dominik Wujastyk. ‘An overview of edmac: a PLAIN TeX format for critical editions’. *TUGboat*, **11**, 4, pp. 623–643, November 1990. (Code available from CTAN in `macros/plain/contrib/edmac`)
- [Lüc03] Uwe Lück. ‘ednotes — critical edition typesetting with LaTeX’. *TUGboat*, **24**, 2, pp. 224–236, 2003. (Code available from CTAN in `macros/latex/contrib/ednotes`)
- [Sul92] Wayne G. Sullivan. *The file edstanza.doc*. June 1992. (Available from CTAN in `macros/plain/contrib/edmac`)
- [Wil02] Peter Wilson. *The memoir class for configurable typesetting*. November 2002. (Available from CTAN in `macros/latex/contrib/memoir`)
- [Wil04] Peter Wilson and Maïeul Rouquette. *Parallel typesetting for critical editions: the eldpar package*. December 2004. (Available from CTAN in `macros/latex/contrib/ledmmac`)

## 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

<code>\#</code>	5712, 5944
<code>\&amp;</code>	38, 4919, 4923, 4924, 4984, 4994, 5016
<code>\@ledleftnote</code>	5092, 5100
<code>\@ledrightnote</code>	5091, 5099
<code>\@ledsidenote</code>	5093, 5101
<code>\@@line</code>	2253
<code>\@EDROWFILL@</code>	5141, <u>5474</u>
<code>\@M</code>	2253, 4979, 4989
<code>\@MM</code>	1967

\@Rlineflag	4251–4253, 4785, 4786, 4814, 4830, 4834
\@addswfalse	1318
\@addswtrue	1320, 1322, 1326
\@adv	<u>743</u> , 1087, 1088
\@afterindenttrue	269
\@apprefprefixmore	<u>4261</u>
\@apprefprefixsingle	<u>4261</u>
\@arabic	1402
\@auxout	4115, 4128, 4183, 4186, 4189, 4192
\@backslashchar	4893
\@checkend	5045
\@currentlabel	1445, 2775, 2935, 3008, 3139
\@currentseries	2510, 2574, 2580, 2589, 2598, 2633, 2635, 2640, 2642, 2645, 2647, 2655, 3354, 3430, 3436, 3445, 3454, 3466, 3468, 3473, 3475, 3478, 3480, 3488, 4293, 4310
\@currenenvir	5029, 5032, 5033
\@doclearpage	219, <u>4062</u>
\@doreinfeetX	3968, 3971, 3972, <u>3986</u>
\@edindex@fornote@true	4720
\@edindex@hyperref	4795, <u>4837</u>
\@edrowfill@	<u>5474</u>
\@edtext@level	964, 988–990, 992– 996, 998, 1000, <u>1156</u> , 1160, 1164, 1166, 1168, 1171, 1173, 1175, 1208, 1256, 1314, 3584, 4173
\@eled@sectioningfalse	1570
\@eled@sectioningtrue	1568
\@emptytoks	<u>5020</u> , 5030
\@first	3750, 3752
\@firstoftwo	1354, 1371
\@firstseries	3756, 3757
\@fnpos	<u>3252</u> , 3936, 3939, 3967, 3970
\@footnotemark	<u>2677</u>
\@footnotetext	<u>2676</u> , <u>2692</u> , 2715, 4604, 4605, 4643, 4670
\@freelist	4026, 4027, 4033, 4034
\@getfirstseries	3754
\@gobble	23, 971, 1147, 1148, 3388, 5042, 5094–5096, 5131, 5708–5710
\@gobblefive	<u>227</u> , 3514, 3702
\@gobblefour	<u>227</u>
\@gobblethree	<u>227</u> , 977, 5706
\@h	<u>2251</u>
\@hangingsymbol	<u>4906</u> , 4915
\@hanginsymbol	4908
\@ifclassloaded	78, 2675, 4019, 4049
\@ifnextchar	5117
\@ifpackageloaded	81, 83, 4023, 4686, 4689, 5875
\@iiiminipage	222, <u>4602</u>
\@index@command	4732, 4742, 4750, 4756, 4758–4760, 4763–4765, 4848, 4852, 4891, 4893
\@index@command@	4759, 4760, 4764, 4765
\@index@parenthesis	4757, 4761, 4766, 4847, 4851, 4872, 4878
\@index@txt	4755, 4791, 4792, 4795, 4802, 4805, 4874, 4880, 4881, 4901
\@indexfile	4801, 4804, 4808
\@inputcheck	590

\@insert ..... 1863–1865, 1899–1901  
\@k ..... 2251  
\@l@dttempcnta ..... 238, 821, 823, 825, 826, 830, 832, 834,  
835, 1641, 1642, 1644, 1646, 1649, 1650, 1665, 1706–1710, 1712, 1719–1723, 1725, 1728,  
1731, 1734, 1739, 1770, 1774, 1778, 1785, 1789, 1793, 1872, 1876, 1880, 1883, 1886, 1889, 1890  
\@l@dttempcntb ..... 238, 427, 429, 431, 438, 442, 446, 450, 453,  
501, 502, 509, 513, 517, 519, 527, 528, 1704, 1716, 1739, 1748–1750, 1752, 1770, 1774,  
1778, 1785, 1789, 1793, 1821–1823, 1825, 1878, 1879, 4332, 4334, 4336, 4342, 4346, 4350,  
4354, 4357, 4444–4446, 4449–4451, 4454, 4462–4464, 4467–4469, 4472, 4523–4525, 4527  
\@lab ..... 975, 4106, 4119, 4161  
\@led@nofootfalse ..... 4065, 4070, 4075  
\@led@nofoottrue ..... 4063  
\@led@testifnofoot ..... 4062  
\@ledinnote@command ..... 4724, 4749, 4791, 4792, 4802, 4805  
\@lemma ..... 976, 1191, 1193, 1255  
\@lemmacommand@false ..... 1161, 1209  
\@lemmacommand@true ..... 1249  
\@line@@num ..... 5120, 5149  
\@lock ..... 291, 570, 679, 681, 683, 696,  
881, 882, 884, 885, 901, 902, 904, 1529, 1611, 1671, 1673, 1674, 1676, 1782, 1797, 1799, 1801  
\@lopL ..... 727, 971  
\@lopR ..... 727  
\@makechapterhead ..... 5759  
\@makecol ..... 213, 4032  
\@makeschapterhead ..... 5759  
\@mem@extranofeet ..... 4050  
\@mem@nofootfalse ..... 4053, 4056  
\@mem@old@ssect ..... 5759  
\@midlist ..... 4026, 4027, 4033, 4034  
\@minipagefalse ..... 4617, 4618  
\@minus ..... 3572, 3646  
\@mpfn ..... 4642, 4669  
\@mpfnpos ..... 3252, 4548, 4551  
\@mpfootins ..... 4623, 4627, 4633, 4635, 4648, 4675  
\@mpfootnotetext ..... 4604, 4605, 4643, 4670  
\@nameuse ..... 534, 536, 2020, 2021, 2071, 2156, 2157, 2202, 2316, 2390, 2742, 2746,  
2748, 2757, 2760, 2761, 2767, 2776, 2780, 2784, 2810, 2815, 2833, 2865, 2871, 2932, 2936,  
2945, 2953, 3005, 3009, 3018, 3026, 3095, 3108, 3116, 3117, 3122, 3140, 3146, 3160, 4575, 4577  
\@new ..... 3737–3740, 3744–3747  
\@next@page ..... 1026, 1027  
\@nl ..... 648, 1027, 1029, 1031, 1038, 1042, 1045  
\@nl@reg ..... 648  
\@nl@regR ..... 652  
\@nbreakfalse ..... 1419, 1567  
\@nbreaktrue ..... 1417, 1421, 1567  
\@noeled@ssectrue ..... 10  
\@noneed@Footnotefalse ..... 1207  
\@noneed@Footnotetrue ..... 3687  
\@oldnbreak ..... 1417, 1419, 1476  
\@opXfeet ..... 3948, 3949

\@outputbox	3946, 3947, 3993, 3994
\@parboxrestore	2025, 2765
\@pend	<u>727</u>
\@pendR	<u>727</u>
\@percentchar	3696
\@plus	3199, 3572, 3646
\@ref	<u>953</u> , 1053, 1062
\@ref@reg	<u>955</u>
\@ref@regR	957
\@reinserts	216, 4040
\@second	3751, 3752
\@secondoftwo	1355, 1372
\@sect	<u>5759</u>
\@series	584, 588, 1979, 2734, 3342, <u>3525</u> , 3527, 3731, 3736, 3739, 3740, 3743, 3745, 3747, 3750, 3751, 3755, 3757, 3770, 3787, 3963, 3982, 4002, 4015, 4059, 4079, 4559, 4582, 4589, 4599, 5078, 5086, 5129, 5134, 5148
\@set	<u>776</u> , 1096, 1097
\@sidenotesepp	<u>4498</u> , 4507
\@ssect	<u>5759</u>
\@startstanza	<u>4981</u>
\@stopstanza	<u>4981</u>
\@sw	977, 1289, 1292, <u>1297</u>
\@tag	<u>1155</u> , <u>1178</u> , 1250, 2701, 2703, 2707, 3601, 3618, 3692
\@tempcnta	1314–1316, 1319, 1325, 1333, 1339, 1343
\@templ@d	4513, 4515
\@templ@n	4514, 4515
\@tmp	995, 997, 998, 1333, 1335, 1337, 1339, 1353–1355, 1370–1372
\@tmpp	996, 997
\@whilenum	1315, 5449
\@wredindex	<u>4784</u>
\@x@sfs	2669, 2672, 2720, 2726
\@xloop	1897, <u>1904</u>
\@xympar	<u>4316</u>
\^	631
\_	4839, 4841, 5482, 5487, 5495

## A

\abs@line@verse	6158, 6159, 6162, 6163
\abs@prevline	6137, 6138
\absline@num	285, <u>569</u> , 659, 662, 665, 724, 813, 828, 847, 869, 891, 922, 932, 949, 966, 1024, 1025, 1034, 1036, 1300, 1301, 1337, 1602, 1623, 1624, 1632, 1862, 6127, 6129, 6133, 6136, 6137, 6145, 6146, 6158, 6162
\absline@numR	810, 819, 840, 919, 929, 946, 1307, 1308, 1335
Abu Kamil Shuja' b. Aslam	11
\actionlines@list	<u>572</u> , 610, 613, 621, 813, 828, 847, 869, 891, 922, 932, 949, 1654, 1657
\actionlines@listR	602, 605, 810, 819, 840, 919, 929, 946
\actions@list	<u>572</u> , 614, 622, 814, 835, 849, 851, 871, 880, 893, 900, 923, 933, 950, 1658
\actions@listR	606, 811, 826, 842, 844, 920, 930, 947

<code>\add@inserts</code>	1551, 1565, <u>1851</u>
<code>\add@inserts@next</code>	<u>1851</u>
<code>\add@penalties</code>	1562, <u>1872</u>
<code>\addcontentsline</code>	5706
<code>\addtocounter</code>	1474
<code>\addtol@denvbody</code>	<u>5024</u> , 5046, 5048
Adelard II	11
<code>\advancelabel@refs</code>	4113, 4126, <u>4134</u>
<code>\advanceline</code>	19, 125, 128, <u>1086</u> , 1129, 1136, 1148
<code>\advancepageno</code>	<u>3931</u>
<code>\Aendnote</code>	22
<code>\affixline@num</code>	1539, <u>1697</u>
<code>\affixpstart@num</code>	1547, <u>1810</u>
<code>\affixside@note</code>	1551, 1565, <u>4500</u>
<code>\Afootnote</code>	22
<code>\afterlemmaseparator</code>	6191
<code>\afternote</code>	6224
<code>\afternoteX</code>	35, 6226
<code>\afternumberinfootnote</code>	6186
<code>\afterruleX</code>	36
<code>\afterXrule</code>	6195
<code>\afterXsymlinenum</code>	6188
<code>\allowbreak</code>	2309, 2379, 2946, 3019
<code>\ampersand</code>	41, <u>4919</u>
<code>\applabel</code>	44, 155, <u>4172</u>
<code>\appref</code>	44, <u>4261</u>
<code>\apprefprefixsingle</code>	4281
<code>\apprefwithpage</code>	44, <u>4261</u>
<code>\appto</code>	4506, 4507
<code>\apptocmd</code>	2676, 2697, 4324, 5777, 5858, 5870
<code>\arrangementX</code>	29, <u>2728</u> , 6170–6172
<code>\arrangementX@normal</code>	2874, <u>2874</u> , 3678
<code>\arrangementX@paragraph</code>	3048
<code>\arrangementX@threecol</code>	<u>2975</u> , <u>3048</u>
<code>\arrangementX@twocol</code>	<u>2901</u>
<code>\at@every@pend</code>	1480, <u>1483</u>
<code>\at@every@pstart</code>	1396, 1397, 1399, 1409
<code>\AtBeginDocument</code>	49, 3212, 4157, 4684, 5759, 6243
<code>\AtEndDocument</code>	53
<code>\AtEveryPend</code>	16, <u>1483</u>
<code>\AtEveryPstart</code>	16, <u>1393</u>
<code>\autopar</code>	15, 146, 337, 1411, 1478, <u>1496</u>
<code>\autopar@pausetrue</code>	333
<code>\autoparfalse</code>	323, 1497
<code>\autopartrue</code>	1510

## B

<code>\ballast</code>	55
<code>\ballast@count</code>	<u>1618</u> , 1621, 1626, 1872
Beeton, Barbara Ann Neuhaus Friend	16

<code>\beforeeledchapter</code>	5714
<code>\beforelemmaseparator</code>	6192
<code>\beforenotesX</code>	36
<code>\beforenumberinfootnote</code>	6185
<code>\beforeXnotes</code>	6205
<code>\beforeXsymlinenum</code>	6187
<code>\beginnumbering</code>	14, 149, <u>252</u> , 354, 1424, 1507
<code>\beginnumberingR</code>	1502
<code>\Bendnote</code>	22
<code>\Bfootnote</code>	22
<code>\bfseries</code>	1402
<code>\bhooknoteX</code>	34
<code>\bhookXendnote</code>	6232
<code>\bhookXnote</code>	6182
<code>\body</code>	1905, 1906, 4921, 5017
<code>\bodyfootmarkA</code>	27
<code>\boolfalse</code>	990, 3331
<code>\booltrue</code>	1256, 3324
<code>\box</code>	1594, 1596, 2151, 2166, 2233, 2252, 3112, 3126
<code>\boxendlinenum</code>	6209
<code>\boxfootnotenumbers</code>	2493, <u>2496</u>
<code>\boxlinenum</code>	6206
<code>\boxlinenumalign</code>	6207
<code>\boxstartlinenum</code>	6208
<code>\boxsymlinenum</code>	6183
<code>\boxXendendlinenum</code>	6236
<code>\boxXendlinenum</code>	6233
<code>\boxXendlinenumalign</code>	6234
<code>\boxXendstartlinenum</code>	6235
Bredon, Simon	11
Breger, Herbert	11, 12, 226
Brey, Gerhard	11
<code>\brokenpenalty</code>	1492
Busard, Hubert L. L.	11
<code>\bypage@false</code>	<u>377</u> , 401, 412
<code>\bypage@true</code>	<u>377</u> , 390
<code>\bypstart@false</code>	<u>377</u> , 391, 413
<code>\bypstart@true</code>	<u>377</u> , 402

## C

<code>\c@addcolcount</code>	<u>5276</u>
<code>\c@ballast</code>	<u>1618</u> , 1626
<code>\c@firstlinenum</code>	<u>459</u> , 1718, 1720, 1723, 1725
<code>\c@firstsublinenum</code>	<u>463</u> , 1705, 1707, 1710, 1712
<code>\c@labidx</code>	4692
<code>\c@linenumincrement</code>	<u>459</u> , 1721, 1722
<code>\c@mpfootnote</code>	4642, 4669
<code>\c@page</code>	1026, 1029, 1031, 1037, 1040, 1042, 1045, 4378, 4386
<code>\c@pstart</code>	1402, 4129, 4190, 4193
<code>\c@pstartR</code>	4116, 4184, 4187

`\c@sublinenumincrement` . . . . . 463, 1708, 1709  
`\Cendnote` . . . . . 22  
`\centerline` . . . . . 5384, 5389, 5395, 5400, 5406, 5411  
`\Cfootnote` . . . . . 22  
`\ch@ck@l@ck` . . . . . 1737, 1766  
`\ch@cksub@l@ck` . . . . . 1714, 1766  
`\changes` . . . . . 4305  
`\chapter` . . . . . 5759, 6055, 6059, 6065, 6066  
`\chaptermark` . . . . . 5708, 6056, 6060  
`\char` . . . . . 4919  
`\chardef` . . . . . 4921, 4923  
`\check@pb@in@verse` . . . . . 1534, 6154  
Chester, Robert of . . . . . 11  
Claassens, Geert H. M. . . . . 11  
`\cleaders` . . . . . 5443  
`\cleardoublepage` . . . . . 5721, 5814, 5817  
`\clearforchapter` . . . . . 5716, 5760, 5763  
`\closeout` . . . . . 53, 325, 1010, 1018, 3333  
`\clubpenalty` . . . . . 1492, 1876  
`\color@begingroup` . . . . . 2026, 2162, 2766, 3122  
`\color@endgroup` . . . . . 2027, 2162, 2767, 3122  
`\columns@position` . . . . . 272, 273, 347, 348, 3219, 3232, 3242, 3248  
`\columnwidth` . . . . . 2024, 2120, 2764, 3076, 3224, 3237, 4657  
`\content` . . . . . 3586,  
3601, 3618, 3663, 3673, 3688, 3693, 4399, 4402, 4406, 4414, 4417, 4421, 4429, 4432, 4436  
Copernicus, Nicolaus . . . . . 11  
`\count` . . . . . 1993,  
2001, 2100, 2109, 2283, 2287, 2358, 2387, 2886, 2894, 2920, 2924, 2994, 2998, 3056, 3065  
`\countdef` . . . . . 3931  
`\cr` . . . . . 2254, 2257  
`\create@this@edtext@level` . . . . . 993, 994  
`\critext` . . . . . 258  
`\cs` . . . . . 4305, 4864, 4865  
`\csdef` . . . . . 4175, 6055, 6056, 6059, 6060, 6065, 6066,  
6074, 6075, 6078, 6079, 6083, 6084, 6092, 6093, 6096, 6097, 6101, 6102, 6109, 6110, 6113, 6114  
`\csgdef` . . . . . 1986, 2094, 2265, 2340, 2875, 2902, 2976, 3049, 3535–3542, 3544, 3545, 3547,  
3548, 3550, 3553, 3554, 3560, 3562, 3563, 3565–3567, 3569–3575, 3577, 3578, 3637–3647,  
3649, 3650, 3704, 3705, 3709–3711, 3713, 3714, 3716, 3717, 3719–3722, 3725, 3726, 3728, 3781  
`\cslet` . . . . . 998, 1339, 3513, 3514, 3568, 3702  
`\csletcs` . . . . . 3632, 4558, 4588, 5072, 5084, 5128, 5133, 5147  
`\csnumdef` . . . . . 1460, 1462  
`\csnumgdef` . . . . . 1288, 1291, 1303, 1310  
`\csundef` . . . . . 583, 1571, 4881  
`\csuse` . . . . . 272, 273, 347, 348, 1288, 1289, 1291,  
1292, 1305, 1312, 1359, 1376, 1569, 1994–1996, 2002–2004, 2010, 2011, 2022, 2023, 2033,  
2036–2038, 2047, 2064, 2090, 2102–2104, 2110–2112, 2116, 2131, 2143, 2144, 2158, 2159,  
2177, 2184–2186, 2194, 2195, 2224, 2225, 2240, 2242–2244, 2246, 2269–2271, 2276, 2277,  
2292, 2297, 2300, 2302, 2305–2307, 2310, 2311, 2334, 2344–2346, 2351, 2352, 2362, 2367,  
2370, 2372, 2375–2377, 2380, 2381, 2408, 2416, 2448, 2454, 2465, 2467–2471, 2492, 2497,  
2501, 2503, 2505, 2518, 2522, 2611, 2633, 2635, 2640, 2642, 2645, 2647, 2655, 2749, 2750,



2757, 2762, 2763, 2779, 2780, 2792, 2845, 2856, 2888–2890, 2895–2897, 2906–2908, 2913,  
 2914, 2929, 2939, 2940, 2944, 2945, 2948, 2980–2982, 2987, 2988, 3003, 3011, 3013, 3017,  
 3018, 3021, 3058–3060, 3066–3068, 3072, 3087, 3104, 3105, 3118, 3119, 3136, 3144, 3145,  
 3156, 3184, 3188, 3189, 3197, 3219, 3232, 3242, 3248, 3257, 3259, 3261, 3265, 3267, 3269,  
 3286, 3287, 3293, 3294, 3307, 3308, 3314, 3315, 3347, 3348, 3359, 3360, 3365, 3374, 3376–  
 3378, 3392, 3393, 3396, 3398, 3400, 3466, 3468, 3473, 3475, 3478, 3480, 3488, 3565, 3566,  
 3600, 3617, 3624, 3665, 3667, 3673, 3764, 3765, 3783, 3784, 3803, 3816, 3950, 3951, 3957–  
 3959, 3979, 3980, 3987, 3988, 3996–3998, 4009, 4011, 4012, 4053, 4056, 4069, 4074, 4252,  
 4253, 4572, 4580, 4596, 4597, 4776, 4779, 4782, 4827, 4880, 5074, 5075, 5738, 5754, 6093, 6097  
`\csxdef` . . . . . 1300, 1307, 1928, 1930, 1934, 1936, 1943,  
 1946, 1949, 1954, 1957, 1960, 1997, 2101, 2121, 2284, 2359, 2484, 2887, 2921, 2995, 3057, 4874  
`\ctab` . . . . . 5055, 5587, 5651  
`\ctabtext` . . . . . 5059, 5597, 5655

## D

`\dcolerr` . . . . . 5196, 5208  
`\dcoli` . . . . . 5166, 5198, 5199, 5486, 5491  
`\dcolii` . . . . . 5167, 5199  
`\dcoliii` . . . . . 5168, 5199  
`\dcoliv` . . . . . 5169, 5200  
`\dcolix` . . . . . 5174, 5201  
`\dcolv` . . . . . 5170, 5200  
`\dcolvi` . . . . . 5171, 5200  
`\dcolvii` . . . . . 5172, 5201  
`\dcolviii` . . . . . 5173, 5201  
`\dcolx` . . . . . 5175, 5201  
`\dcolxi` . . . . . 5176, 5202  
`\dcolxii` . . . . . 5177, 5202  
`\dcolxiii` . . . . . 5178, 5202  
`\dcolxiv` . . . . . 5179, 5203  
`\dcolxix` . . . . . 5184, 5204  
`\dcolxv` . . . . . 5180, 5203  
`\dcolxvi` . . . . . 5181, 5203  
`\dcolxvii` . . . . . 5182, 5204  
`\dcolxviii` . . . . . 5183, 5204  
`\dcolxx` . . . . . 5185, 5204  
`\dcolxxi` . . . . . 5186, 5205  
`\dcolxxii` . . . . . 5187, 5205  
`\dcolxxiii` . . . . . 5188, 5205  
`\dcolxxiv` . . . . . 5189, 5206  
`\dcolxxix` . . . . . 5194, 5207  
`\dcolxxv` . . . . . 5190, 5206  
`\dcolxxvi` . . . . . 5191, 5206  
`\dcolxxvii` . . . . . 5192, 5207  
`\dcolxxviii` . . . . . 5193, 5207  
`\dcolxxx` . . . . . 5195, 5207  
`\DeclareOptionX` . . . . . 6, 10, 13, 16, 19, 22, 30, 34, 35, 39, 42, 45, 48, 57, 62  
`\default@series` . . . . . 6, 3734  
Dekker, Dirk-Jan . . . . . 56  
`\Dendnote` . . . . . 22

<code>\detokenize</code>	1281
<code>\Dfootnote</code>	22
<code>\dimen</code>	1073, 1074, 1078–1080, 1084, 1994, 2002, 2102, 2110, 2118–2120, 2122, 2259–2261, 2269, 2285, 2288, 2344, 2360, 2388, 2888, 2895, 2906, 2922, 2925, 2980, 2996, 2999, 3058, 3066, 3074–3076, 3079, 3257, 3265
<code>\dimexpr</code>	2047, 2131, 2792, 3087, 4953
<code>\dingdef</code>	4575, 4577, 4633, 4635
<code>\disable@familiarnotes</code>	5069, 5105
<code>\disable@notes</code>	5103, 5288, 5307, 5323, 5339, 5354, 5371, 5560
<code>\disable@sidenotes</code>	5090, 5104
<code>\disablenl@dtabfeet</code>	5154, 5286, 5305, 5321, 5337, 5352, 5369, 5493, 5500, 5505, 5513, 5518, 5526, 5544, 5561
<code>\displaystyle</code>	5226, 5289, 5292, 5324, 5327, 5355, 5358, 5493, 5505, 5518, 5608, 5633, 5634
<code>\displaywidowpenalty</code>	1493
<code>\divide</code>	1708, 1721, 2120, 2260, 3076
<code>\do@actions</code>	1603, 1630
<code>\do@actions@fixedcode</code>	1651, 1664
<code>\do@actions@next</code>	1630
<code>\do@ballast</code>	1604, 1618
<code>\do@feetX</code>	3937, 3940, 3941, 3986
<code>\do@insidelinehook</code>	1549, 1581, 1584
<code>\do@line</code>	1463, 1519
<code>\do@linehook</code>	1523, 1581, 1583
<code>\do@lockoff</code>	890
<code>\do@lockoffL</code>	890
<code>\do@lockoffR</code>	910
<code>\do@lockon</code>	854
<code>\do@lockonL</code>	854
<code>\do@lockonR</code>	861
<code>\docsvlist</code>	1329, 1924, 1982, 2737, 3358, 3385, 3523, 3772, 3776, 3789, 3793, 3806, 3819
<code>\doedindexlabel</code>	4697, 4819
<code>\doendnotes</code>	3497
<code>\doendnotesbysection</code>	22, 3506
<code>\doinsidelinehook</code>	20, 1583
<code>\dolinehook</code>	20, 1583
<code>\dolistloop</code>	584, 588, 1979, 2734, 3342, 3770, 3787, 3963, 3982, 4002, 4015, 4059, 4079, 4511, 4531, 4538, 4559, 4582, 4589, 4599, 5078, 5086, 5129, 5134, 5148
<code>\dosplits</code>	2251
Downes, Michael	55, 136, 138
<code>\doxtrafeet</code>	3935
<code>\dp</code>	1969, 2149, 2164, 3110, 3124
<code>\dummy@edtext</code>	1141, 1150, 6056, 6060, 6075, 6079, 6093, 6097
<code>\dummy@edtext@showlemma</code>	1142, 5772, 5853, 5865, 6055, 6059, 6065, 6066
<code>\dummy@ref</code>	954, 970

## E

<code>\edaftertab</code>	51, 237, 5061, 5533, 5569, 5587
<code>edarrayc</code> (environment)	48, 5650
<code>edarrayl</code> (environment)	48, 5650
<code>edarrayr</code> (environment)	48, 5650

<code>\edatleft</code>	50, 5062, <u>5423</u>
<code>\edatright</code>	50, 5063, <u>5431</u>
<code>\edbeforetab</code>	51, 237, 5060, <u>5533</u> , <u>5569</u> , <u>5587</u>
<code>\edfilldimen</code>	<u>5273</u> , 5450, 5451, 5453, <u>5476</u>
<code>\edfont@info</code>	1228, 1231, 1237, 1240, <u>1245</u>
<code>\EDINDEX</code>	<u>5115</u>
<code>\edindex</code>	46, <u>4784</u> , 5115, 5130, 5135, 5146, 5279, 5298, 5316, 5332, 5347, 5364
<code>\edindexlab</code>	48, <u>4692</u> , 4699, 4705, 4732, 4750, 4849, 4853, 4858, 4860, 4890
<code>\EDLABEL</code>	<u>5113</u>
<code>\edlabel</code>	42, 1147, <u>4100</u> , 4699, 5113, 5137, 5152, 5278, 5297, 5315, 5331, 5346, 5363, 5492, 5499, 5504, 5512, 5517, 5525
<code>\edlineref</code>	42, <u>4208</u> , 6244
<code>\edmakelabel</code>	44, <u>4314</u>
<code>\edpageref</code>	42, <u>4205</u>
<code>\edrowfill</code>	49, 5066, 5138, 5141, <u>5474</u>
<code>\EDTAB</code>	<u>5632</u>
<code>\edtabcolsep</code>	49, <u>5255</u> , 5293, 5312, 5327, 5343, 5359, 5376, 5451, 5507, 5520, 5529, 5618
<code>\EDTABINDENT</code>	<u>5613</u> , 5626
<code>\edtabindent</code>	5481, 5485, 5490, 5501, 5514, 5527, <u>5622</u>
<code>\EDTABtext</code>	<u>5640</u>
<code>edtabularc (environment)</code>	48, <u>5654</u>
<code>edtabularl (environment)</code>	48, <u>5654</u>
<code>edtabularr (environment)</code>	48, <u>5654</u>
<code>\EDTEXT</code>	<u>5111</u>
<code>\edtext</code>	21, 98, 155, 1150, <u>1158</u> , 2694, 2869, 3599, 4374, 4375, 4393, 5111, 5127, 5145, 5771, 5772, 5778, 5852, 5853, 5859, 5864, 5865, 5871, 6055, 6056, 6059, 6060, 6065, 6066, 6075, 6079, 6093, 6097
<code>\edvertdots</code>	51, 5065, <u>5442</u>
<code>\edvertline</code>	51, 5064, <u>5440</u>
<code>\Eendnote</code>	22
<code>\Efootnote</code>	22
<code>\eled@chapter</code>	5951, 5955, 6003, 6007, <u>6050</u>
<code>\eled@section</code>	5964, 5968, 6016, 6020, <u>6050</u>
<code>\eled@sectioning@out</code>	279, 325, <u>5946</u> , 5954, 5967, 5980, 5992, 6006, 6019, 6032, 6045
<code>\eled@sectioningR@out</code>	5950, 5963, 5976, 5988, 6002, 6015, 6028, 6041
<code>\eled@sections@@</code>	276, 1541, 1567
<code>\eled@subsection</code>	5977, 5981, 6029, 6033, <u>6050</u>
<code>\eled@subsubsection</code>	5989, 5993, 6042, 6046, <u>6050</u>
<code>\eledchapter</code>	<u>5947</u>
<code>\eledchapter*</code>	<u>5947</u>
<code>\eledmac@xindy@out</code>	49, 50, 53, 4728, 4738, 4884
<code>\eledmaccompat@true</code>	63
<code>\eledmacmarkuplocdepth</code>	52, 4731, 4741, 4887
<code>\eledsection</code>	5817, <u>5947</u>
<code>\eledsection*</code>	<u>5947</u>
<code>\eledsubsection</code>	<u>5947</u>
<code>\eledsubsection*</code>	<u>5947</u>
<code>\eledsubsubsection</code>	<u>5947</u>
<code>\eledsubsubsection*</code>	<u>5947</u>
<code>\emph</code>	4783

`\empty` . 236, 250, 310, 313, 359, 362, 602, 610, 1165, 1167, 1172, 1174, 1198, 1226, 1235, 1262,  
 1266, 1272, 1349, 1366, 1431, 1654, 1717, 1733, 1853–1855, 1866, 1898, 4107, 4120, 4934  
`\enablel@dtabfeet` . . . . . 5154, 5541, 5557, 5577, 5585, 5595, 5603  
`\end@lemmas` . . . . . 1140, 1198, 1199  
`\endashchar` . . . . . 38, 2413, 2650, 3483  
`\endgraf` . . . . . 1458, 1512, 1516  
`\endline@num` . . . . . 577, 980, 986  
`\endlock` . . . . . 19, 1110, 1146, 4988, 5001, 5009  
`\endminipage` . . . . . 225, 4608  
`\endnumbering` . . . . . 14, 149, 255, 300, 334, 353  
`\endpage@num` . . . . . 577, 979, 986  
`\endprint` . . . . . 3345, 3502, 3513  
`\endquotation` . . . . . 5658  
`\endquote` . . . . . 5658  
`\endsub` . . . . . 19, 1071, 1145  
`\endsubline@num` . . . . . 577, 981, 987  
`\enlargethispage` . . . . . 6139, 6147  
`\enspace` . . . . . 3403  
 environments:  
   `edarrayc` . . . . . 48, 5650  
   `edarrayl` . . . . . 48, 5650  
   `edarrayr` . . . . . 48, 5650  
   `edtabularc` . . . . . 48, 5654  
   `edtabularl` . . . . . 48, 5654  
   `edtabularr` . . . . . 48, 5654  
   `ledgroup` . . . . . 42, 4640  
   `ledgroupsize` . . . . . 42, 4654  
   `minipage` . . . . . 42  
 Euclid . . . . . 11  
`\ExecuteOptionsX` . . . . . 7, 36  
`\expandonce` . . . . . 997, 1949, 1960,  
 2703, 2707, 2846, 2857, 3601, 3618, 3673, 3692, 3693, 4402, 4406, 4417, 4421, 4432, 4436, 4784  
`\extensionchars` . . . . . 54, 248, 261, 341, 641, 643  
`\extractendline@` . . . . . 2439, 2442  
`\extractendsubline@` . . . . . 2440, 2442  
`\extractline@` . . . . . 2437, 2442, 2445  
`\extractsubline@` . . . . . 2438, 2442, 2445

## F

`\f@encoding` . . . . . 1245  
`\f@family` . . . . . 1245  
`\f@series` . . . . . 1245  
`\f@shape` . . . . . 1245  
`\f@x@l@cks` . . . . . 1537, 1760, 1766  
 Fairbairns, Robin . . . . . 27  
`\first@linenum@out@false` . . . . . 1004, 1011  
`\first@linenum@out@true` . . . . . 1004  
`\firstlinenum` . . . . . 18, 468  
`\firstseriesX@` . . . . . 3301, 3301, 3306, 3309, 3312, 3314, 3992  
`\firstsublinenum` . . . . . 18, 468

\firstXseries@	3280, <u>3280</u> , 3285, 3288, 3291, 3293, 3295, 3316, 3955
\fix@page	650, <u>701</u>
\flag@end	<u>1048</u> , 1187, 1204, 1205, 1215
\flag@start	<u>1048</u> , 1186, 1187, 1205
\flagstanza	41, <u>5013</u>
\floatingpenalty	1967
\flush@notes	1468, <u>1896</u> , 4568
\flush@notesR	4566
\fnpos	28, <u>3252</u>
Folkerts, Menso	11
\fontencoding	1910
\fontfamily	1910
\fontseries	1910
\fontshape	1910
\footfootmarkA	27
\footfudgefiddle	55, <u>2115</u> , 2120, 3076
\footins	4064, 4090
\footnormal	6176
\footnormalX	6170
\footnote@luatexpardir	2772, 3195, 3671
\footnote@luatextextdir	2418, 2771, 3194, 3670
\footnoteA	27
\footnoteB	27
\footnoteC	27
\footnoteD	27
\footnoteE	27
\footnotelang@lua	<u>1926</u> , 2840, 2851, 3590, 3607
\footnotelang@poly	<u>1940</u> , 2844, 2855, 3594, 3611
\footnoteoptions@	<u>1913</u> , 3595, 3603, 3612, 3621
\footnoterule	1972, 2807, 4610, 4611, 4625
\footnotesize	3541, 3643, 3710, 4371, 4372
\footnoteXnomk	3666
\footparagraph	6177
\footparagraphX	6171
\footsplitskips	<u>1964</u> , 2012, 2145, 2160, 2293, 2363, 2751, 2930, 3004, 3106, 3120
\footthreecol	6179
\footthreecolX	6173
\foottwocol	6178
\foottwocolX	6172
\fulllines@	<u>3926</u>
\fullstop	38, 553, <u>2413</u> , 2636, 2638, 2651, 2653, 3471, 3486

## G

\g@addto@macro	4020, 4021, 4050
Gädeke, Nora	11
\get@edindex@hyperref	4794, <u>4837</u>
\get@edindex@ledinnote@command	<u>4722</u> , 4790, 4800
\get@index@command	<u>4754</u> , 4789, 4799, 4845, 4856, 4869, 4870
\get@linelistfile	598, 626
\get@sw@txt	<u>1277</u> , 1286, 1298, 1347

\getline@num ..... 1527, 1601  
 \gl@p ..... 374, 605, 606,  
     613, 614, 1168, 1175, 1199, 1230, 1239, 1353, 1370, 1657, 1658, 1859, 1863, 1899, 4110, 4123  
 \gl@poff ..... 374, 375

## H

\h@num ..... 936  
 \hangafter ..... 4977  
 \hangindentX ..... 34  
 \hb@xt@ ..... 1548, 1553, 1594, 1596, 5476, 5481, 5485, 5490, 5501, 5514, 5527, 5606, 5608  
 \hfilneg ..... 2253  
 \hide@num ..... 939, 941, 944  
 \hidenumbering ..... 20, 936  
 \Hilfsbox ..... 5161  
 \hilfsbox ..... 5151, 5161, 5216, 5217, 5226, 5238, 5289, 5308,  
     5324, 5340, 5355, 5372, 5492, 5494, 5499, 5503, 5504, 5506, 5512, 5516, 5517, 5519, 5525, 5528  
 \hilfscount ..... 5161, 5617–5619, 5625  
 \HILFSskip ..... 5610  
 \Hilfsskip ..... 5482, 5486, 5487, 5491, 5494, 5495, 5502,  
     5503, 5506–5508, 5515, 5516, 5519–5521, 5528–5530, 5610, 5616, 5618, 5624, 5628, 5629  
 \hilfsskip ..... 5150, 5151, 5161, 5292, 5311, 5327, 5343, 5358, 5375, 5627–5629  
 \hsize@fornote ..... 3215, 3220, 3223, 3224, 3228, 3233, 3236, 3237  
 \hsizethreecol ..... 6181  
 \hsizethreecolX ..... 35  
 \hsizetwocol ..... 6180  
 \hsizetwocolX ..... 35  
 \Hy@raisedlink ..... 4117, 4130, 4185, 4191  
 \hyperlink ..... 4201, 4202, 4776, 4826, 4827, 4830, 4888  
 \hyperlinkformat ..... 4824, 4834  
 \hyperlinkformatR ..... 4833  
 \hyperlinkR ..... 4829  
 \hyperpage ..... 4782  
 \hypertarget ..... 4117, 4130, 4185, 4191

## I

\if@addsw ..... 1296, 1332  
 \if@edindex@fornote@ ..... 4717, 4788, 4798  
 \if@edindex@fornote@true ..... 4717  
 \if@eled@sectioning ..... 5727, 5729, 5744, 5761, 5815, 5835  
 \if@firstcolumn ..... 1742, 1815, 4517  
 \if@led@nofoot ..... 4047, 4091  
 \if@lemmacommand@ ..... 1189, 1259  
 \if@nobreak ..... 1416  
 \if@noeled@sec ..... 4, 277, 324  
 \if@noneed@Footnote ..... 1048  
 \if@openright ..... 5720, 5814, 5817  
 \if@RTL ..... 88, 1187, 1205, 1572, 1942, 1953, 2151, 5731, 5747, 5834, 5836  
 \ifautopar ..... 333, 1410, 1435, 1477, 1496, 4101, 5663, 5676, 5685, 5698  
 \ifautopar@pause ..... 337, 1518  
 \IfBeginWith ..... 4758, 4763

\ifbool	3322, 3338
\ifboolexpr	2426, 2511, 2574, 2578, 3361, 3370, 3430, 3434, 4770
\ifbypage@	<u>377</u> , 718, 1635, 2549, 2616
\ifbypage@R	<u>379</u> , 708
\ifbypstart@	<u>377</u> , 1464
\ifbypstart@R	<u>379</u>
\ifcsdef	587, 1316, 1357, 1374, 2241, 2459, 2610
\ifcseempty	1166, 1173, 2036, 2184, 2243, 2305, 2375, 2463, 2574, 2598, 3372, 3430, 3454
\ifcsequal	2461
\ifcsstring	2191, 2192, 2221, 2222, 3152, 3153, 3180, 3181
\ifcsundef	994, 1164, 1171, 1302, 1309, 4174
\ifdef	88, 4117, 4130, 4185, 4191, 4201, 6244
\ifdefempty	3755, 4294, 4891
\ifdefequal	4771, 4843
\ifdefined	3529, 4251
\ifdefstring	2418
\ifdim	1074, 1078, 1080, 1084, 2668, 5216, 5617
\ifdimequal	2042, 2126, 2468, 2497, 2633, 2640, 2655, 2787, 3082, 3282, 3303, 3393, 3466, 3473, 3488
\ifeledmaccompat@	<u>4</u> , 6168
\IfEq	4297
\iffirst@linenum@out@	<u>1004</u> , 1009
\ifhbox	2232, 2237
\ifhmode	2719, 2726
\ifHy@hyperindex	4771, 4843
\ifinserthangingsymbol	<u>4911</u> , 6155
\ifinstanza	1124, 1131, 1437, 1513, <u>4906</u> , 4914, 6155
\ifistwofollowinglines@	2600, 2608, 3456
\ifl@d@dash	<u>2524</u> , 2573, 2650, 3429, 3483
\ifl@d@elin	<u>2524</u> , 2567, 2652, 2653, 3423, 3485, 3486
\ifl@d@esl	<u>2524</u> , 2653, 3486
\ifl@d@pnum	<u>2524</u> , 2555, 2636, 2651, 3411, 3484
\ifl@d@ssub	<u>2524</u> , 2638, 3471
\ifl@d@Xmorethantwolines	<u>2524</u> , 2644, 3477
\ifl@d@Xtwolines	<u>2524</u> , 2643, 3476
\ifl@dend@X	<u>3681</u>
\ifl@dhidenummer	<u>936</u> , 1535
\ifl@dmemoir	<u>77</u> , 5714
\ifl@dpaging	239, 2013, 2752
\ifl@dpairing	<u>239</u> , 264, 307, 327, 343, 2013, 2053, 2072, 2078, 2203, 2209, 2317, 2323, 2391, 2397, 2752, 2799, 2816, 2822, 2954, 2960, 3027, 3033, 3161, 3168, 4101, 4564, 4573, 4631
\ifl@dprintingcolumns	<u>239</u>
\ifl@dprintingpages	<u>239</u> , 1966, 5762, 5816
\ifl@dskipnumber	<u>1119</u> , 1700
\ifl@dskipversenumber	<u>1119</u> , 1740
\ifl@dstartendok	5447, <u>5457</u>
\ifl@imakeidx	<u>80</u> , 4685, 4787
\ifl@indextools	<u>82</u> , 4688
\iflabelpstart	1405, 1445
\ifledfinal	<u>4</u> , 54, 230

\ifledgroupnotesL@ . . . . . 1698, 4682  
 \ifledgroupnotesR@ . . . . . 4682  
 \iflednopbinverse . . . . . 4, 6154, 6155  
 \ifledRcol . . . . . 239, 428, 470, 477,  
     484, 491, 592, 600, 640, 651, 705, 746, 760, 778, 796, 809, 818, 839, 860, 909, 918, 928,  
     938, 945, 956, 1051, 1075, 1081, 1087, 1096, 1106, 1112, 1116, 1123, 1163, 1182, 1190,  
     1216, 1225, 1287, 1299, 1334, 1499, 2427, 2702, 2838, 3588, 4105, 4163, 4182, 4333, 4401,  
     4416, 4431, 4565, 4574, 4632, 4846, 4857, 4889, 5949, 5962, 5975, 5987, 6001, 6014, 6027, 6040  
 \ifledRcol@ . . . . . 169, 176, 183, 239, 1348, 4443, 4461  
 \ifleftnoteup . . . . . 4483, 4496  
 \ifluatex . . . . . 547, 1413, 1554, 2417, 2629, 2770, 2839, 2850, 3193, 3589, 3606, 3669  
 \ifnocritical@ . . . . . 4, 392, 403, 414, 1247, 3532, 3832, 3954, 3977, 4052, 4068, 4557, 4563, 6175  
 \ifnoend@ . . . . . 4, 3322, 3680, 3898, 6229  
 \ifnofamiliar@ . . . . . 4, 3635, 3881, 3991, 4007, 4055, 4073, 4587, 4594, 5070, 5082, 6216  
 \ifnoledgroup@ . . . . . 4, 1998, 2069, 2106, 2273, 2348, 2891, 2910, 2984, 3062, 3580, 3656, 4545  
 \ifnoquotation@ . . . . . 4, 5659  
 \ifnoteschanged@ . . . . . 317, 581  
 \ifnumberedpar@ . . . . .  
     . . 1159, 1388, 1426, 1454, 2693, 2700, 2837, 2868, 3587, 3691, 3692, 4317, 4400, 4415, 4430  
 \ifnumbering . . . . . 251, 253, 301, 336, 382, 1422, 1450, 1505  
 \ifnumberingR . . . . . 239, 1500  
 \ifnumberline . . . . . 1222, 1605, 1699  
 \ifnumberpstart . . . . . 1403, 1436, 1471, 1513  
 \ifnumequal . . . . . 1325, 1465, 2240, 2242, 2588, 2614, 2617, 2619, 2620, 3188, 3444, 3512, 4504  
 \ifnumgreater . . . . . 1362, 1379, 3752, 4512, 4532, 4539  
 \ifodd . . . . . 1752, 1825, 4378, 4386, 4454, 4472, 4527  
 \ifparapparatus@ . . . . . 4  
 \ifparledgroup . . . . . 4, 2072, 2077,  
     2203, 2208, 2317, 2322, 2391, 2396, 2816, 2821, 2954, 2959, 3027, 3032, 3161, 3167, 4573, 4630  
 \ifpst@rtedL . . . . . 239  
 \ifpstartnum . . . . . 1836, 1839, 1844  
 \ifrightnoteup . . . . . 4394, 4491  
 \ifseriesbefore . . . . . 3291, 3312, 3749  
 \ifsidepstartnum . . . . . 1438, 1810  
 \ifstrempty . . . . . 1395, 1409, 1480, 1485, 1977, 2732, 3769, 3771, 3786,  
     3788, 3800, 3813, 4825, 6051, 6053, 6064, 6071, 6072, 6082, 6089, 6090, 6100, 6107, 6108, 6112  
 \IfStrEq . . . . . 638, 1023, 1033, 1321, 1528, 1544,  
     3285, 3306, 3936, 3939, 3967, 3970, 4290, 4307, 4548, 4551, 4872, 4878, 6135, 6144, 6157, 6161  
 \IfStrEqCase . . . . . 1045  
 \ifstrequal . . . . . 1915, 1927, 1941, 3762, 3782  
 \ifsublines@ . . . . . 551, 568, 691, 732,  
     737, 745, 779, 786, 820, 829, 841, 848, 870, 892, 985, 987, 1606, 1643, 1703, 4139, 4165, 4169  
 \IfSubStr . . . . . 2503, 2505, 3398, 3400, 4789, 4799, 4844, 4868  
 \iftoggle . . . . . 2036, 2044, 2128,  
     2184, 2305, 2375, 2441, 2447, 2452, 2457, 2515, 2521, 2789, 3084, 3217, 3230, 3241, 3247, 3350  
 \iftrue . . . . . 4771, 4843  
 \ifvbox . . . . . 1461, 4041, 4042  
 \ifvmode . . . . . 4112, 4125  
 \ifvoid . . . . . 3957, 3979, 3996, 4009, 4053, 4056, 4064, 4069, 4074, 4090, 4572, 4596, 4648, 4675  
 \ifwidthliketwocolumns . . . . . 4, 271, 346



<code>\ifXendinsertsep@</code>	3346, <a href="#">3497</a>
<code>\ifxetex</code>	1278
<code>\ifxindy@</code>	<a href="#">4</a> , 4723
<code>\ifxindyhyperref@</code>	<a href="#">4</a> , 4727, 4867
<code>\imki@wrindexentry</code>	86, 4791, 4792, 4795
<code>\indexentry</code>	4802, 4805, 4809
<code>\indtl@wrindexentry</code>	86
<code>\initnumbering@quote</code>	270, 345, <a href="#">5658</a>
<code>\initnumbering@reg</code>	<a href="#">252</a>
<code>\inplaceoflemmaseparator</code>	6193
<code>\inplaceofnumber</code>	6189
<code>\InputIfFileExists</code>	278, 627
<code>\insert</code>	2009, 2141, 2291, 2361, 2748, 2928, 3002, 3102, 3980, 4010
<code>\insert@count</code>	<a href="#">952</a> , <a href="#">953</a> , 1062, 1064, 1183, 1917, 1929, 1931, 1944, 1947, 1950, 2709, 2859, 3620, 4408, 4423, 4438
<code>\insert@countR</code>	.. 1053, 1055, 1182, 1921, 1935, 1937, 1955, 1958, 1961, 2705, 2848, 3604, 4404, 4419, 4434
<code>\inserthangingsymbol</code>	1557, 4912
<code>\inserthangingsymbolfalse</code>	1532
<code>\inserthangingsymboltrue</code>	1530
<code>\inserthangingsymbol</code>	<a href="#">4911</a>
<code>\insertlines@list</code>	310, <a href="#">572</a> , 620, 966, 1855, 1859
<code>\insertparafootsepX</code>	3142, 3187
<code>\inserts@list</code>	1430, <a href="#">1850</a> , 1853, 1863, 1898, 1899, 1916, 1928, 1930, 1943, 1946, 1949, 2708, 2858, 3619, 4407, 4422, 4437
<code>\inserts@listR</code>	1920, 1934, 1936, 1954, 1957, 1960, 2704, 2847, 3602, 4403, 4418, 4433
<code>\instanzafalse</code>	5004
<code>\instanzatrue</code>	4982
<code>\interfootnotelinepenalty</code>	1965
<code>\interlinepenalty</code>	1493, 1883, 1965, 4988
<code>\istwofollowinglines@false</code>	2613
<code>\istwofollowinglines@true</code>	2615, 2620
<code>\itemcount@</code>	170, 172, 177, 179, 184, 186, 4502, 4504, 4509, 4512, 4530, 4532, 4537, 4539

## J

Jayaditya	11
-----------	----

## K

Kabelschacht, Alois	127
---------------------	-----

## L

<code>\l@advance@parledgroup@beforenormalnotes</code>	4611, 4622
<code>\l@d@add</code>	1267, 1269, 1273, <a href="#">1275</a>
<code>\l@d@dashfalse</code>	2548, 2594, 3406, 3450
<code>\l@d@dashtrue</code>	2552, 2558, 2570, 3409, 3414, 3426
<code>\l@d@elinfalse</code>	2555, 2596, 3411, 3452
<code>\l@d@elintrue</code>	2555, 2557, 3411, 3413
<code>\l@d@err@UnequalColumns</code>	5267
<code>\l@d@eslfalse</code>	2564, 2567, 2597, 3420, 3423, 3453
<code>\l@d@esltrue</code>	2567, 2569, 3423, 3425

<code>\l@d@index</code>	5117
<code>\l@d@nums</code>	1181, 1228, 1231, 1237, 1240, 1261, 1262, 1275, 3598, 3601, 3615, 3618, 3691, 4180
<code>\l@d@pnumfalse</code>	2548, 3406
<code>\l@d@pnumtrue</code>	2551, 3408
<code>\l@d@section</code>	3340, 3345, 3511
<code>\l@d@set</code>	794, 1106, 1107
<code>\l@d@ssubfalse</code>	2560, 3416
<code>\l@d@ssubtrue</code>	2562, 3418
<code>\l@d@Xend</code>	3681
<code>\l@d@Xmorethantwolinesttrue</code>	2601, 3457
<code>\l@d@Xtwolinesttrue</code>	2595, 3451
<code>\l@d@ampcount</code>	5156, 5263, 5265, 5270, 5490, 5500, 5501, 5513, 5514, 5549, 5567, 5605
<code>\l@d@begin@stack</code>	5030, 5041–5043
<code>\l@d@bfnote</code>	2694, 2698
<code>\l@d@checkcols</code>	5222, 5234, 5260
<code>\l@d@checkstartend</code>	5446, 5457
<code>\l@d@chset@num</code>	658, 661, 794
<code>\l@d@colcount</code>	5156, 5198, 5210, 5211, 5221, 5223, 5233, 5235, 5261, 5265, 5270, 5280, 5282, 5299, 5301, 5317, 5318, 5333, 5334, 5348, 5349, 5365, 5366, 5418, 5419, 5490, 5500, 5501, 5513, 5514, 5545, 5547, 5562, 5564, 5605, 5614, 5623
<code>\l@d@collect@body</code>	5032, 5040
<code>\l@d@collect@body</code>	5027, 5650–5652, 5654–5656
<code>\l@d@colwidth</code>	5150, 5198, 5216, 5217, 5417, 5476, 5502, 5515, 5606, 5608, 5616, 5617
<code>\l@d@csnote</code>	4393, 4394
<code>\l@d@csnotetext</code>	1586, 4457, 4473, 4478, 4511, 4514
<code>\l@d@csnotetext@l</code>	1586, 4455, 4514, 4531
<code>\l@d@csnotetext@r</code>	1586, 4475, 4514, 4538
<code>\l@d@dodoreinextrafeet</code>	3966, 4021, 4042
<code>\l@d@dextrafeet</code>	3935, 4020, 4027, 4034
<code>\l@d@dedbeginmini</code>	4546, 4556
<code>\l@d@dedendmini</code>	4549, 4552, 4553, 4556
<code>\l@d@emptyd@ta</code>	1524, 1586
<code>\l@d@dend@close</code>	24, 3323, 3499, 3507
<code>\l@d@dend@open</code>	23, 3323, 3339
<code>\l@d@dend@stuff</code>	25, 262, 342, 3336
<code>\l@d@dend@Xfalse</code>	3681
<code>\l@d@dend@Xtrue</code>	3681
<code>\l@d@envbody</code>	5022, 5025, 5028–5030
<code>\l@d@fambeginmini</code>	4546, 4586
<code>\l@d@famendmini</code>	4549, 4552, 4553, 4586
<code>\l@d@feetbeginmini</code>	4545, 4605, 4644, 4671
<code>\l@d@feetendmini</code>	4545, 4618, 4651, 4678
<code>\l@d@getline@margin</code>	423
<code>\l@d@getlock@disp</code>	498, 526
<code>\l@d@getref@num</code>	4205, 4206, 4208, 4209, 4211, 4212, 4214, 4215, 4222, 4244, 4249
<code>\l@d@getsidenote@margin</code>	4329
<code>\l@d@gobblearg</code>	5121
<code>\l@d@gobbleoptarg</code>	5121
<code>\l@d@hidenumberfalse</code>	1536
<code>\l@d@hidenumbertrue</code>	1691

\l@dlabell@parse	4228, 4231
\l@dld@ta	1550, <u>1586</u> , <u>1741</u> , 1816, 1828, 5735, 5751, 5754
\l@dldp@rbox	1594, <u>4364</u> , 4482
\l@dlsn@te	1552, <u>1593</u> , 5736, 5752, 5754
\l@dlsnote	4374, <u>4394</u>
\l@dmake@labels	4129, <u>4149</u> , 4158, 4190, 4193
\l@dmake@labelsR	4116, 4184, 4187
\l@dmemoirfalse	78
\l@dmemoirtrue	78
\l@dmodforedtext	5126, 5154
\l@dnnullfills	<u>5136</u> , 5534, 5552, 5570, 5580, 5588, 5598
\l@dnumpstartsL	239, 265, 284, 304, 328, 1433, 1541, 1566, 1569, 1571, 5955, 5968, 5981, 5993, 6007, 6020, 6033, 6046
\l@dnumpstartsR	5951, 5964, 5977, 5989, 6003, 6016, 6029, 6042
\l@doldold@footnotetext	2692
\l@dp@rsefootspec	<u>2531</u> , 4180, 4719
\l@dpagingfalse	239
\l@dpagingtrue	239
\l@dpairingfalse	239
\l@dpairingtrue	239
\l@dparsedendline	<u>2531</u> , 4187, 4193, 4715
\l@dparsedendpage	<u>2531</u> , 4187, 4193, 4713
\l@dparsedendsub	<u>2531</u> , 4187, 4193
\l@dparsedstartline	<u>2531</u> , 4184, 4190, 4710
\l@dparsedstartpage	<u>2531</u> , 4184, 4190, 4708
\l@dparsedstartsub	<u>2531</u> , 4184, 4190
\l@dparsefootspec	<u>2531</u>
\l@dprintingcolumnsfalse	239
\l@dprintingcolumnstrue	239
\l@dprintingpagesfalse	239
\l@dprintingpagestrue	239
\l@dpush@begins	<u>5037</u> , 5041
\l@drd@ta	1559, <u>1586</u> , <u>1741</u> , 1818, 1826, 5735, 5738, 5751
\l@dref@undefined	4205, 4208, 4211, 4214, 4217
\l@drestorefills	<u>5136</u> , 5538, 5554, 5574, 5582, 5592, 5600
\l@dstoreforedtext	5126, 5155
\l@drp@rbox	1596, <u>4364</u> , 4490
\l@drrsn@te	1560, <u>1593</u> , 5736, 5738, 5752
\l@drrsnote	4375, <u>4394</u>
\l@dsetmaxcolwidth	<u>5215</u> , 5228, 5240
\l@dskipnumberfalse	<u>1119</u> , 1701
\l@dskipnumbertrue	<u>1119</u> , 1687
\l@dskipversenumberfalse	1453
\l@dskipversenumbertrue	1689
\l@dstartendokfalse	5461, 5465, 5469
\l@dstartendoktrue	5459
\l@dtabaddcols	<u>5445</u> , 5475
\l@dtabnoexpands	1151, <u>5053</u>
\l@dunboxmpfoot	<u>4622</u> , 4649, 4676
\l@dunhbox@line	<u>1519</u> , 1558, 1574, 1577

<code>\l@deropenalties</code>	1448, 1457, <u>1491</u>
<code>\l@makeidxtrue</code>	81, 85
<code>\l@indextoolstrue</code>	84
<code>\l@luatexttextdir@L</code>	1414, 1555
<code>\l@prev@nopb</code>	288, 1024, 1035, 6122, 6129, 6130, 6136, 6145
<code>\l@prev@pb</code>	287, 6121, 6127, 6128, 6133
<code>\label</code>	44
<code>\label@refs</code>	4108, 4110, 4116, 4121, 4123, 4129, 4137, 4140, 4142, 4144
<code>\labelpstartfalse</code>	<u>1393</u>
<code>\labelpstarttrue</code>	16, <u>1393</u>
<code>\labelref@list</code>	<u>4097</u> , 4120, 4123, 4170
<code>\labelref@listR</code>	4107, 4110, 4166
<code>\labelrefsparseline</code>	<u>4134</u>
<code>\labelrefsparsesubline</code>	<u>4134</u>
<code>\language</code>	1949, 1960
<code>\last@page@num</code>	<u>701</u> , 6156
<code>\last@page@numR</code>	706, 712
<code>\lastbox</code>	1511, 1526, 2172, 2231, 2236, 3131
<code>\lastkern</code>	2668
<code>\lastskip</code>	1073, 1079
<code>\latex@makecol</code>	4025
Lavagnino, John	10
<code>\ldots</code>	98
<code>\led@check@nopb</code>	1528, 1544, <u>6133</u>
<code>\led@check@pb</code>	1528, 1544, <u>6133</u>
<code>\led@err@AutoparNotNumbered</code>	<u>134</u> , 1501, 1506
<code>\led@err@Edtextoutsidepstart</code>	<u>97</u> , 1211
<code>\led@err@EdtextWithoutFootnote</code>	<u>200</u> , 1057, 1066
<code>\led@err@FootnoteWithoutEdtext</code>	<u>203</u> , 3628
<code>\led@err@HighEndColumn</code>	<u>189</u> , 5466
<code>\led@err@LineationInNumbered</code>	<u>109</u> , 383
<code>\led@err@LowStartColumn</code>	<u>189</u> , 5462
<code>\led@err@ManyLeftnotes</code>	168, 4532
<code>\led@err@ManyRightnotes</code>	168, 4539
<code>\led@err@ManySidenotes</code>	168, 4512
<code>\led@err@NumberingNotStarted</code>	<u>91</u> , 321
<code>\led@err@NumberingShouldHaveStarted</code>	<u>91</u> , 352
<code>\led@err@NumberingStarted</code>	<u>91</u> , 254
<code>\led@err@NumberingWithoutPstart</code>	<u>134</u> , 305
<code>\led@err@PendNoPstart</code>	<u>134</u> , 1455
<code>\led@err@PendNotNumbered</code>	<u>134</u> , 1451
<code>\led@err@PstartInPstart</code>	<u>134</u> , 1427
<code>\led@err@PstartNotNumbered</code>	<u>134</u> , 1423
<code>\led@err@ReverseColumns</code>	<u>189</u> , 5470
<code>\led@err@TooManyColumns</code>	<u>189</u> , 5212
<code>\led@err@UnequalColumns</code>	<u>189</u>
<code>\led@error@fail@patch@doclearpage</code>	<u>218</u> , 4086, 4093
<code>\led@error@fail@patch@iiiminipage</code>	<u>221</u> , 4607
<code>\led@error@fail@patch@makecol</code>	<u>212</u> , 4029, 4036
<code>\led@error@fail@patch@reinserts</code>	<u>215</u> , 4044

\led@error@fail@patch@endminipage	224, 4613, 4620
\led@error@ImakeidxAfterEledmac	206, 4686
\led@error@IndextoolsAfterEledmac	209
\led@error@indextoolsAfterEledmac	4689
\led@mess@NotesChanged	99, 318
\led@mess@SectionContinued	107, 340
\led@nopb	6125, 6127
\led@nopbnum	6126, 6127
\led@pb	6123, 6127
\led@pb@setting	1023, 1033, 1045, 1528, 1544, 6131, 6135, 6144, 6157, 6161
\led@pbnum	6124, 6127
\led@toksa	364, 372
\led@toksb	364, 371–373
\led@warn@AppLabelOutEdtext	152, 4197
\led@warn@BadAction	150, 1693
\led@warn@BadAdvancelineLine	124, 763, 769
\led@warn@BadAdvancelineSubline	124, 749, 755
\led@warn@BadLineation	112, 418
\led@warn@BadLinenummargin	112, 452
\led@warn@BadLockdisp	112, 504
\led@warn@BadSetline	130, 1094
\led@warn@BadSetlinenum	130, 1104
\led@warn@BadSidenotemargin	161, 4356
\led@warn@BadSublockdisp	112, 530
\led@warn@DuplicateLabel	152, 4152, 4178
\led@warn@LineFileObsolete	122
\led@warn@NoIndexFile	163
\led@warn@NoLineFile	120, 632
\led@warn@NoMarginpars	159, 4318
\led@warn@Obsolete	122, 641, 643
\led@warn@RefUndefined	152, 4219
\led@warn@SeriesStillExist	165, 3527
\ledfinalfalse	35
\ledfinaltrue	34
ledgroup (environment)	42, 4640
ledgroupsized (environment)	42, 4654
\ledinnernote	45, 4374
\ledinnote	4732, 4742, 4769
\ledinnotehyperpage	4769
\ledinnotemark	4769
\ledleftnote	45, 4374, 5092, 5095, 5100
\ledlinenum	542
\ledllfill	1553, 1598, 4658, 4662
\ledlsnotefontsetup	46, 4367, 4481
\ledlsnotesep	46, 1594, 4367
\ledlsnotewidth	46, 4367, 4481
\lednopb	53, 6123
\lednopbinversetrue	42, 53
\lednopbnum	6123, 6163
\ledouternote	45, 4385

<code>\ledouterote</code>	4374
<code>\ledpb</code>	53, <u>6123</u>
<code>\ledpbnum</code>	<u>6123</u> , 6159
<code>\ledpbsetting</code>	53, <u>6131</u>
<code>\ledrightnote</code>	45, <u>4374</u> , 5091, 5094, 5099
<code>\ledrlfill</code>	1559, <u>1598</u> , 4659, 4666
<code>\ledrsnotefontsetup</code>	46, <u>4367</u> , 4489
<code>\ledrsnotesep</code>	46, 1596, <u>4367</u>
<code>\ledrsnotewidth</code>	46, <u>4367</u> , 4489
<code>\ledsectnomark</code>	5707
<code>\ledsectnotoc</code>	5706
<code>\ledsetnormalparstuff@common</code>	3192
<code>\ledsetnormalparstuffX</code>	2778, 3143, <u>3192</u>
<code>\ledsidenote</code>	45, <u>4374</u> , 5093, 5096, 5101
<code>\left</code>	5425, 5428, 5433, 5436
<code>\leftctab</code>	5489, 5589
<code>\leftlinenum</code>	19, <u>542</u> , 1743, 1755, 5733, 5749
<code>\leftlinenumR</code>	5734, 5750
<code>\leftltab</code>	<u>5480</u> , 5571
<code>\leftmargin</code>	5668, 5669, 5690, 5691
<code>\leftnoteupfalse</code>	46
<code>\leftnoteuptrue</code>	4497
<code>\leftpstartnum</code>	<u>1810</u>
<code>\leftrtab</code>	<u>5484</u> , 5535
Leibniz	11
<code>\lemma</code>	23, <u>1247</u>
<code>\lemmaseparator</code>	6190
<code>\letcs</code>	995, 996, 1333, 4255, 4256
<code>\letsforverteilen</code>	<u>5144</u> , 5291, 5310, 5326, 5342, 5357, 5374
<code>\line@list</code>	313, <u>572</u> , 619, 987, 1235, 1239
<code>\line@list@stuff</code>	261, 341, <u>1007</u>
<code>\line@list@version</code>	<u>637</u> , 1013
<code>\line@listR</code>	1226, 1230
<code>\line@margin</code>	<u>423</u> , 1748, 1821
<code>\line@marginR</code>	429
<code>\line@num</code>	172, 179, 186, 289, 550, <u>566</u> , 663, 697, 719, 767, 768, 770, 789, 801, 802, 832, 980, 984, 1612, 1636, 1646, 1716, 1718, 1719, 1728, 1729, 2242, 4168
<code>\line@numR</code>	170, 177, 184, 709, 761, 762, 764, 782, 797, 798, 823, 4164
<code>\line@set</code>	1263, <u>1264</u>
<code>\lineation</code>	18, 110, 113, <u>381</u>
<code>\lineinfo@</code>	2442, 2445, 2484
<code>\linenum</code>	23, <u>1260</u> , 4258, 5120, 5131, 5149
<code>\linenum@out</code>	941, <u>1003</u> , 1010, 1012, 1013, 1018, 1019, 1027, 1029, 1031, 1038, 1040, 1042, 1045, 1061, 1076, 1082, 1088, 1097, 1107, 1113, 1117, 1132, 1134, 1193, 1219, 1289, 4119, 6123–6126
<code>\linenum@outR</code>	939, 1052, 1075, 1081, 1087, 1096, 1106, 1112, 1116, 1125, 1127, 1191, 1217, 1292, 4106
<code>\linenumberlist</code>	18, <u>236</u> , 1717, 1729
<code>\linenumberstyle</code>	20, <u>533</u>
<code>\linenumincrement</code>	18, <u>468</u>
<code>\linenummargin</code>	18, 115, <u>423</u>

<code>\linenumr@p</code>	533, 4164, 4168
<code>\linenumrep</code>	533, 550, 2637, 2652, 3470, 3485
<code>\linenumsep</code>	19, 542, 1840, 1845, 4369, 4370
<code>\lineref</code>	6244
<code>\linesinpar@listL</code>	623
<code>\linesonpage@listL</code>	624
<code>\linewidth</code>	1548
<code>\list@clear</code>	361, 597, 619–624, 1430
<code>\list@clearing@reg</code>	595, 618
<code>\list@clearing@regR</code>	593
<code>\list@create</code>	358, 572–575, 988, 993, 1140, 1850, 4097
<code>\listbreak</code>	1326
<code>\listcsgadd</code>	6068, 6086, 6104, 6116
<code>\listead</code>	3738, 3739, 3745, 3746
<code>\listgadd</code>	4455, 4457, 4473, 4475, 4478
<code>\listxadd</code>	724, 3731, 6127–6130
<code>\lock@disp</code>	498, 1784, 1788, 1792
<code>\lock@off</code>	857, 858, 890, 1116, 1117
<code>\lock@on</code>	854, 1112, 1113
<code>\lockdisp</code>	19, 117, 498
Lorch, Richard	11
<code>\ltab</code>	5056, 5569, 5650
<code>\ltabtext</code>	5058, 5579, 5654
<code>\luatexpardir</code>	1930, 1936, 2772, 3195, 3671
<code>\luatextextdir</code>	548, 1414, 1555, 1928, 1934, 2630, 2771, 3194, 3670
Luecking, Dan	59

## M

<code>\m@mdodoreinextrafeet</code>	4021
<code>\m@mdoextrafeet</code>	4020
<code>\m@mmf@check</code>	2667, 2680, 2721
<code>\m@mmf@prepare</code>	2664, 2676, 2687, 2725, 3673
<code>\M@sect</code>	5759
<code>\m@th</code>	5443
<code>\makehboxofhboxes</code>	2193, 2223, 2228, 3154, 3182
<code>\managestanza@modulo</code>	4943, 4971
<code>\marginparwidth</code>	4367, 4368
<code>\marks</code>	2073– 2075, 2204–2206, 2318–2320, 2392–2394, 2817–2819, 2955–2957, 3028–3030, 3163–3165
<code>\mathchardef</code>	4935
<code>\maxdimen</code>	2146, 2161, 3107, 3121
<code>\maxhnotesX</code>	37
<code>\maxhXnotes</code>	6204
<code>\maxlinesinpar@list</code>	597
Mayer, Gyula	11
<code>\measurebody</code>	5537, 5543, 5573, 5591
<code>\measurecell</code>	5220, 5246
<code>\measuremrow</code>	5244, 5548
<code>\measuretbody</code>	5553, 5559, 5581, 5599
<code>\measuretbody</code>	5232, 5251

<code>\measuretrow</code>	5249, 5565
<code>\message</code>	108, 260
Middleton, Thomas	11, 76
<code>minipage</code> (environment)	42
Mittelbach, Frank	10, 11
<code>\morenoexpands</code>	55, 1143
<code>\morethantwolines</code>	6211
<code>\moveleft</code>	3259, 3267, 5482, 5487, 5495
<code>\moveright</code>	5508, 5521, 5530
<code>\mpfnpos</code>	28, 3252
<code>\mpnormalfootgroup</code>	2000, 2069
<code>\mpnormalfootgroupX</code>	2814, 2893
<code>\mpnormalvfootnote</code>	1999, 2019, 2274, 2349
<code>\mpnormalvfootnoteX</code>	2759, 2892, 2911, 2985
<code>\mppara@footgroupX</code>	3064, 3150
<code>\mppara@vfootnoteX</code>	3063, 3101
<code>\mpparafootgroup</code>	2108, 2200
<code>\mpparavfootnote</code>	2107, 2155
<code>\mpthreecolfootgroup</code>	2275, 2315
<code>\mpthreecolfootgroupX</code>	2986, 3021
<code>\mpthreecolfootsetup</code>	2278, 2286
<code>\mpthreecolfootsetupX</code>	2989, 2993
<code>\mptwocolfootgroup</code>	2350, 2386
<code>\mptwocolfootgroupX</code>	2912, 2948
<code>\mptwocolfootsetup</code>	2353, 2386
<code>\mptwocolfootsetupX</code>	2915, 2919
<code>\multfootsep</code>	27, 2661, 2671
<code>\multiplefootnotemarker</code>	2661, 2665, 2666, 2668

## N

<code>\n@l</code>	1040
<code>\n@num</code>	917, 1127, 1134
<code>\n@num@stanza</code>	927, 1125, 1132
<code>\nc@page</code>	1037, 1038
<code>\NeedsTeXFormat</code>	2
<code>\new@line</code>	1022, 1553, 1574, 1577
<code>\newbox</code>	1388, 1391, 4364, 4365, 5161, 5163, 5647, 5648
<code>\newcommandx</code>	1284, 1346, 1407, 1450, 1913, 1926, 1940, 1973, 2728, 3684, 3759, 3779, 3796, 3809, 3810, 3823, 3930, 4289, 4306, 4769, 4784, 4964, 4997, 4999, 5007, 6219, 6224
<code>\newcounter</code>	459, 461, 463, 465, 1401, 1619, 3676, 4134, 4135, 4694, 4943, 5276
<code>\newhookcommand@series</code>	3796, 3834, 3835, 3838–3856, 3869, 3871, 3872, 3874, 3875, 3877, 3878, 3883–3889, 3894– 3896, 3899, 3900, 3903, 3904, 3906, 3907, 3909, 3910, 3912, 3914–3917, 3920, 3921, 3923
<code>\newhookcommand@series@reload</code>	3827, 3857, 3866, 3867, 3890, 3891, 3893
<code>\newhooktoggle@series</code>	3809, 3833, 3836, 3837, 3858–3865, 3882, 3901, 3902, 3913, 3919, 3924
<code>\newhooktoggle@series@reload</code>	3822, 3868, 3892
<code>\newif</code>	5, 9, 12, 15, 18, 21, 29, 33, 38, 41, 44, 47, 56, 61, 77, 80, 82, 88, 239–243, 245–247, 251, 377–380, 568, 581, 936, 1004, 1048, 1119, 1120, 1222, 1259, 1296, 1389, 1403, 1405, 1496, 1518, 1811, 1836, 2524–2530, 2608, 3497, 3682, 4047, 4394, 4496, 4682, 4683, 4717, 4910, 4911, 5457, 5727



\newinsert	3579, 3581, 3655, 3657
\newlength	542, 4927
\newlinechar	3686
\newread	590
\newrobustcmd	4200
\newseries	3521, 3733, 3734
\newseries@	3522, <u>3526</u>
\newseries@par	3529, 3530
\newtoggle	3272, 3276, 3533, 3534, 3551, 3552, 3555–3559, 3561, 3564, 3576, 3636, 3648, 3706–3708, 3724, 3729, 3926–3929, 4263, 4264, 4268, 4269
\newverse	4981
\newwrite	50, 1003, 3681, 5946
\NEXT	<u>5123</u> , 5221, 5224, 5229, 5230, 5233, 5236, 5241, 5242, 5245, 5247, 5248, 5250, 5252, 5253, <u>5380</u> , 5383, 5385, 5386, 5388, 5390, 5391, 5394, 5396, 5397, 5399, 5401, 5402, 5405, 5407, 5408, 5410, 5412, 5413, 5418, 5420, 5421, 5614, 5620, 5621
\Next	5281, 5283, 5294, 5295, 5300, 5302, 5313, 5314, 5317, 5319, 5328, 5329, 5333, 5335, 5344, 5345, 5348, 5350, 5360, 5361, 5365, 5367, 5377, 5378
\next@absline	1034, 1035
\next@action	151, 614, 1625, 1633, 1634, 1640, 1641, 1649, 1658
\next@actionline	611, 613, 1624, 1632, 1655, 1657
\next@actionlineR	603, 605
\next@actionR	606
\next@insert	1431, 1854, 1857, 1859, 1862, 1866
\next@page@num	294, 666, 668, 723, 814
\next@page@numR	713, 811
\no@expands	<u>1143</u> , 1179, 1251
\noalign	2256
\nocritical@true	13
\noeledsec	52
\noend@true	26
\nofamiliar@true	16
\noindent	1397, 1409, 1480, 1487, 1513, 2062, 2064, 2139, 2147, 2151, 2162, 2196, 2226, 2311, 2334, 2381, 2408, 3108, 3122, 3157, 3185, 3203, 3208
\nointerlineskip	3258, 3260, 3266, 3268
\noledgroup@true	19
\nomk@	<u>3929</u>
\nonbreakableafternumber	6203
\nonum@	<u>3927</u>
\nonumberinfootnote	6198
\noquotation@true	30
\normal@footnotemarkX	<u>2740</u> , 2878
\normal@page@break	286, 724, 1025, 1036, <u>6120</u> , 6138, 6146
\normal@pars	303, 1408, 1432, 1475, <u>1516</u> , 2296, 2366, 2938, 3012, 3198
\normalbfnoteX	<u>2835</u> , 2869
\normalbodyfootmarkX	<u>2745</u> , 2879
\normalcolor	2076, 2207, 2321, 2395, 2820, 2958, 3031, 3166, 4624
\normalfont	544, 2662, 2746, 3540, 3642, 3709
\normalfootfmt	1989, <u>2029</u>
\normalfootfmtX	<u>2769</u> , 2882
\normalfootfootmarkX	<u>2783</u> , 2883

<code>\normalfootgroup</code>	1990, <a href="#">2063</a>
<code>\normalfootgroupX</code>	<a href="#">2809</a> , <a href="#">2884</a>
<code>\normalfootnoterule</code>	<a href="#">1972</a> , <a href="#">1992</a>
<code>\normalfootnoteruleX</code>	<a href="#">2807</a> , <a href="#">2885</a> , <a href="#">3055</a>
<code>\normalfootstart</code>	<a href="#">1987</a> , <a href="#">2041</a>
<code>\normalfootstartX</code>	<a href="#">2786</a> , <a href="#">2876</a>
<code>\normalvfootnote</code>	<a href="#">1988</a> , <a href="#">2008</a>
<code>\normalvfootnoteX</code>	<a href="#">2747</a> , <a href="#">2880</a>
<code>\nosep@</code>	<a href="#">3928</a>
<code>\notblank</code>	<a href="#">1924</a> , <a href="#">3358</a> , <a href="#">3385</a>
<code>\notbool</code>	<a href="#">1319</a> , <a href="#">2008</a> , <a href="#">2019</a> , <a href="#">2031</a> , <a href="#">2290</a> , <a href="#">2295</a> , <a href="#">2361</a> , <a href="#">2365</a> , <a href="#">2747</a> , <a href="#">2769</a> , <a href="#">2927</a> , <a href="#">2934</a> , <a href="#">3001</a> , <a href="#">3007</a> , <a href="#">3345</a> , <a href="#">3583</a> , <a href="#">5713</a> , <a href="#">5945</a>
<code>\notefontsizeX</code>	<a href="#">34</a>
<code>\notenumfontX</code>	<a href="#">33</a>
<code>\noteschanged@false</code>	<a href="#">581</a> , <a href="#">628</a>
<code>\noteschanged@true</code>	<a href="#">311</a> , <a href="#">314</a> , <a href="#">581</a> , <a href="#">633</a> , <a href="#">1227</a> , <a href="#">1236</a> , <a href="#">1856</a>
<code>\noteswidthliketwocolumnsX</code>	<a href="#">37</a> , <a href="#">6217</a>
<code>\notesXwidthliketwocolumns</code>	<a href="#">6217</a>
<code>\nottoggle</code>	<a href="#">2035</a> , <a href="#">2183</a> , <a href="#">2298</a> , <a href="#">2304</a> , <a href="#">2368</a> , <a href="#">2374</a> , <a href="#">2941</a> , <a href="#">3014</a> , <a href="#">3203</a> , <a href="#">3208</a> , <a href="#">3367</a> , <a href="#">3381</a> , <a href="#">3666</a>
<code>\NR@ssect</code>	<a href="#">5877</a> , <a href="#">5885</a>
<code>\NR@sssect</code>	<a href="#">5893</a> , <a href="#">5901</a>
<code>\nulledindex</code>	<a href="#">5115</a> , <a href="#">5130</a> , <a href="#">5279</a> , <a href="#">5298</a> , <a href="#">5316</a> , <a href="#">5332</a> , <a href="#">5347</a> , <a href="#">5364</a>
<code>\nullsetzen</code>	<a href="#">5415</a> , <a href="#">5546</a> , <a href="#">5563</a>
<code>\num@lines</code>	<a href="#">1388</a> , <a href="#">1458</a> , <a href="#">1873</a> , <a href="#">1879</a> , <a href="#">1882</a>
<code>\numberedpar@false</code>	<a href="#">1388</a>
<code>\numberedpar@true</code>	<a href="#">1388</a> , <a href="#">1444</a>
<code>\numberingfalse</code>	<a href="#">251</a> , <a href="#">302</a>
<code>\numberingtrue</code>	<a href="#">251</a> , <a href="#">257</a> , <a href="#">334</a>
<code>\numberlinefalse</code>	<a href="#">17</a>
<code>\numberlinetrue</code>	<a href="#">17</a> , <a href="#">1223</a>
<code>\numberonlyfirstinline</code>	<a href="#">6196</a>
<code>\numberonlyfirstintwoline</code>	<a href="#">6197</a>
<code>\numberpstartfalse</code>	<a href="#">16</a> , <a href="#">1393</a>
<code>\numberpstarttrue</code>	<a href="#">16</a> , <a href="#">1393</a>
<code>\numdef</code>	<a href="#">1034</a> , <a href="#">1301</a> , <a href="#">1305</a> , <a href="#">1308</a> , <a href="#">1312</a> , <a href="#">1358</a> , <a href="#">1359</a> , <a href="#">1361</a> , <a href="#">1375</a> , <a href="#">1376</a> , <a href="#">1378</a> , <a href="#">1566</a> , <a href="#">2611</a> , <a href="#">2612</a> , <a href="#">6137</a>
<code>\numgdef</code>	<a href="#">1026</a> , <a href="#">1037</a> , <a href="#">4502</a> , <a href="#">4509</a> , <a href="#">4530</a> , <a href="#">4537</a> , <a href="#">6158</a> , <a href="#">6162</a>
<code>\numlabfont</code>	<a href="#">37</a> , <a href="#">542</a>

## O

<code>\old@edtext</code>	<a href="#">5771</a> , <a href="#">5778</a> , <a href="#">5852</a> , <a href="#">5859</a> , <a href="#">5864</a> , <a href="#">5871</a>
<code>\old@hsize</code>	<a href="#">2054</a> , <a href="#">2066</a> , <a href="#">2197</a> , <a href="#">2800</a> , <a href="#">2811</a> , <a href="#">3210</a>
<code>\old@Rlineflag</code>	<a href="#">4785</a> , <a href="#">4814</a>
<code>\one@line</code>	<a href="#">1388</a> , <a href="#">1525</a> , <a href="#">1526</a> , <a href="#">1553</a> , <a href="#">1558</a> , <a href="#">1574</a> , <a href="#">1577</a>
<code>\onlyXpstart</code>	<a href="#">6201</a>
<code>\openout</code>	<a href="#">49</a> , <a href="#">279</a> , <a href="#">1012</a> , <a href="#">1019</a> , <a href="#">3326</a>

## P

<code>\p@pstart</code>	<a href="#">1446</a>
<code>\PackageError</code>	<a href="#">90</a>
<code>\PackageWarning</code>	<a href="#">89</a>

\page@action	667, <u>807</u> , 972
\page@num	172, 179, 186, <u>577</u> , 609, 721, 979, 984, 1634, 1750, 1823, 2240, 2249, 3188, 4378, 4386, 4451, 4469, 4525, 6156
\page@numR	170, 177, 184, 601, 711, 4446, 4464
\pagebreak	6133
\pagelinesep	47, <u>4692</u> , 4704, 4709, 4714
\pageno	153, 155, 157, <u>3931</u>
\pageref	44
\pairs	6057, 6060, 6075, 6079, 6093, 6097
\paperwidth	1573, 1576
\par@line	<u>1388</u> , 1459, 1874, 1875, 1878, 1882
\para@footgroupX	3054, <u>3150</u>
\para@footsetup	2105, <u>2116</u>
\para@footsetupX	3061, <u>3072</u>
\para@vfootnoteX	3052, <u>3101</u>
\parafootfmt	2098, <u>2179</u>
\parafootfmtX	3053, <u>3138</u>
\parafootgroup	2099, <u>2189</u>
\parafootsep	6219
\parafootsepX	35, 6221
\parafootstart	2096, <u>2124</u>
\parafootstartX	3051, <u>3081</u>
\parapparatus@true	39
\paravfootnote	2097, <u>2140</u>
\parindentX	34
\parledgroup@	2073, 2204, 2318, 2392, 2817, 2955, 3028, 3163
\parledgroup@beforenotesL	4577, 4635
\parledgroup@beforenotesR	4575, 4633
\parledgroup@series	2074, 2205, 2319, 2393, 2818, 2956, 3029, 3164
\parledgroup@type	2075, 2206, 2320, 2394, 2819, 2957, 3030, 3165
\parpackage	3666
\patchcmd	2678, 2684, 4024, 4031, 4039, 4088, 4602, 4608, 4615, 5760, 5782, 5790, 5798, 5806, 5814, 5824, 5833, 5842, 5877, 5885, 5893, 5901, 5910, 5918, 5926, 5934
\pausenumbering	17, <u>332</u>
\pend	15, 98, 141, 144, 1428, <u>1450</u> , 1514, 5002, 5009
Plato of Tivoli	11
\postbodyfootmark	<u>2717</u> , 2743
\postdisplaypenalty	1494
\prebodyfootmark	<u>2717</u> , 2741
\predisplaypenalty	1493
\prenotesX	36, 3279
\prenotesX@	2787, 3082, 3278, 3279, 3303, 3307
\prepare@edindex@fornote	3598, 3615, <u>4718</u>
\prepare@prenotesX	<u>3301</u> , 3302, 3662
\prepare@preXnotes	3280, 3281, 3597, 3614
\pretocmd	2692, 4082, 4316, 4817, 5770, 5851, 5863
\prev@line	1301–1303, 1305, 1308–1310, 1312, 1358, 1359, 1375, 1376
\prev@nopb	<u>6121</u>
\prev@pb	<u>6121</u>
\prevgraf	1458

<code>\prevline</code>	2241, 2460
<code>\prevpage@num</code>	<u>2239</u> , <u>3187</u>
<code>\preXnotes</code>	36, <u>3272</u> , <u>3276</u>
<code>\preXnotes@</code>	2042, 2126, <u>3272</u> , <u>3276</u> , 3282, 3286
<code>\print@eledsection</code>	1542, <u>1564</u>
<code>\print@footnoteXrule</code>	2804, 2826, 2831, 2964, 2969, 3037, 3042, 3098, 3172, 3177, <u>3256</u>
<code>\print@leftmargin@eledsection</code>	<u>5728</u> , 5793, 5801, 5826, 5844, 5888, 5904, 5921, 5937
<code>\print@line</code>	1543, <u>1546</u>
<code>\print@notesX</code>	<u>3986</u>
<code>\print@rightmargin@eledsection</code>	<u>5728</u> , 5785, 5809, 5828, 5846, 5880, 5896, 5913, 5929
<code>\print@Xfootnoterule</code>	2061, 2082, 2087, 2138, 2213, 2218, 2327, 2332, 2401, 2406, <u>3256</u>
<code>\print@Xnotes</code>	<u>3949</u>
<code>\printendlines</code>	3394, 3399, <u>3464</u> , 4311
<code>\printlineendnotearea</code>	3366, <u>3390</u>
<code>\printlinefootnote</code>	2034, 2182, 2303, 2373, <u>2436</u>
<code>\printlinefootnotearea</code>	2474, 2478, 2482, <u>2490</u>
<code>\printlinefootnotenumbers</code>	2498, 2504, <u>2509</u>
<code>\printlines</code>	2515, <u>2627</u> , 4301
<code>\printnpnum</code>	3469, 3484, <u>3495</u>
<code>\printpstart</code>	<u>2425</u> , 2514
<code>\printXafternumber</code>	2494, 2520
<code>\printXbeforenumber</code>	2491, <u>2517</u>
<code>\processl@denvbody</code>	5029, 5033, 5034, 5050
<code>\ProcessOptionsX</code>	65
<code>\protected@csxdef</code>	3665, 5074
<code>\protected@edef</code>	1445, 2775, 2935, 3008, 3139
<code>\protected@write</code>	1289, 1292, 4115, 4128, 4183, 4186, 4189, 4192, 4801, 4804, 4808
<code>\protected@xdef</code>	2845, 2856
<code>\providebool</code>	989
<code>\ProvidesPackage</code>	3
<code>\pst@rtedLfalse</code>	<u>239</u> , 266, 283, 308
<code>\pst@rtedLtrue</code>	<u>239</u> , 338
<code>\pstart</code>	15, 98, 135, 138, 139, 144, 149, <u>1393</u> , 1513, 4973
<code>\pstarteref</code>	<u>4214</u>
<code>\pstartinfootnote</code>	6199
<code>\pstartinfootnoteeverytime</code>	6200
<code>\pstartline</code>	1462, 1465
<code>\pstartnum</code>	<u>1810</u>
<code>\pstartnumfalse</code>	1841, 1848
<code>\pstartnumtrue</code>	1472, 1837
<code>\pstartref</code>	42, <u>4214</u>

## Q

<code>\quotation</code>	5658
<code>\quote</code>	<u>5658</u>

## R

<code>\RaggedLeft</code>	2191, 2221, 3152, 3180
<code>\RaggedRight</code>	2192, 2222, 3153, 3181
<code>\raggedright</code>	3539, 3641, 4372, 5834, 5836

\raggedX	36
\raw@text	<u>1388</u> , 1434, 1461, 1525
\rbracket	38, <u>2413</u> , 3568
\read@linelist	<u>590</u> , 1008
\ref	44
\Relax	<u>5123</u>
\reledmac@error	<u>90</u> , 92, 94, 96, 98, 110, 135, 138, 141, 144, 146, 149, 190, 192, 195, 197, 199, 201, 204, 207, 210, 213, 216, 219, 222, 225
\reledmac@warning	<u>89</u> , 113, 115, 117, 119, 121, 123, 125, 128, 131, 133, 151, 153, 155, 157, 160, 162, 164, 166, 170, 172, 177, 179, 184, 186
\rem@inder	1729, 1731–1733
\removehboxes	2194, 2224, <u>2228</u> , 3155, 3183
\removelastskip	5280, 5299
\RequirePackage	4, 67–69, 71–76
\reserveinserts	70
\resetprevline@	79, 296, <u>582</u> , 1465, 1637
\resetprevpage@	<u>586</u>
\resetprevpage@num	80, 297, 586, 4641
\restore@familiarnotes	<u>5069</u> , 5109
\restore@notes	<u>5103</u> , 5290, 5309, 5325, 5341, 5356, 5373, 5566
\restore@sidenotes	<u>5090</u> , 5108
\resumenumbering	17, <u>332</u>
\right	5426, 5429, 5434, 5437
\rightctab	<u>5498</u> , 5590
\rightlinenum	19, <u>542</u> , 1745, 1753, 5733, 5749
\rightlinenumR	5734, 5750
\rightltab	<u>5511</u> , 5572
\rightnoteupfalse	46
\rightnoteuptrue	4395
\rightpstartnum	1818, 1826, 1843
\rightrtab	<u>5524</u> , 5536
\rightstartnum	<u>1810</u>
\rigidbalance	<u>2251</u> , 2314, 2337, 2384, 2411, 2951, 2973, 3024, 3046
\rlap	1745, 1753, 1818, 1826, 5732, 5748
\roman	5277
\rtab	5054, <u>5533</u> , 5652
\rtabtext	5057, <u>5551</u> , 5656

## S

Sacrobosco	11
\sameword	25, 1149, <u>1284</u>
\sameword@inedtext	1149, <u>1346</u>
\sc@n@list	1730, 1732
Schöpf, Rainer	11
\section@num	 .. <u>248</u> , 258, 260, 261, 278, 279, 339–341, 641, 643, 1300, 1302, 1303, 1305, 1357, 1359, 3340
\section@numR	1307, 1309, 1310, 1312, 1374, 1376
\sectionmark	5709, 6075, 6079
\select@lemmfont	1144, <u>1908</u>
\select@lemmfont	38, <u>1908</u> , 2035, 2183, 2304, 2374, 3368

<code>\series</code>	3521
<code>\seriesatbegin</code>	28, 3735
<code>\seriesatend</code>	28, 3742
<code>\set@line</code>	1181, 1224
<code>\set@line@action</code>	660, 774, 784, 791, 816, 974
<code>\setapprefprefixmore</code>	44, 4261
<code>\setapprefprefixsingle</code>	44, 4261
<code>\setcommand@series</code>	3779, 3798, 3829
<code>\sethangingsymbol</code>	40, 41, 4906
<code>\setistwofollowinglines</code>	2577, 2608, 3433
<code>\setl@dlp@rbox</code>	4480, 4518, 4533, 4535
<code>\setl@drp@rbox</code>	4488, 4520, 4528, 4540
<code>\setl@drpr@box</code>	4480
<code>\setline</code>	19, 131, 1091, 1148, 1465
<code>\setlinenum</code>	20, 133, 1101
<code>\setmcellcenter</code>	5346, 5406
<code>\setmcellleft</code>	5315, 5395
<code>\setmcellright</code>	5278, 5384
<code>\setmrowcenter</code>	5404, 5594
<code>\setmrowleft</code>	5393, 5576
<code>\setmrowright</code>	5382, 5540
<code>\setnoteswidthliketwocolumnsX@</code>	2753, 2802, 2824, 2829, 2962, 2967, 3035, 3040, 3073, 3096, 3159, 3170, 3175, 3210
<code>\setnotesXpositionliketwocolumns@</code>	2755, 2803, 2825, 2830, 2963, 2968, 3036, 3041, 3097, 3171, 3176, 3240
<code>\setprintendlines</code>	3405, 3465
<code>\setprintlines</code>	2547, 2632
<code>\setsidenotessep</code>	46, 4498
<code>\setstanzaindents</code>	39, 4940
<code>\setstanzapenalties</code>	40, 4940
<code>\setstanzavalues</code>	4930, 4940, 4941, 5018
<code>\settccllcenter</code>	5363, 5411
<code>\settccllleft</code>	5331, 5400
<code>\settccllright</code>	5297, 5389
<code>\settoggle</code>	1916, 1920, 3761
<code>\settoggle@series</code>	3759, 3811, 3824
<code>\setthrowcenter</code>	5409, 5602
<code>\setthrowleft</code>	5398, 5584
<code>\setthrowright</code>	5387, 5556
<code>\setXnotespositionliketwocolumns@</code>	2016, 2057, 2081, 2086, 2137, 2212, 2217, 2326, 2331, 2400, 2405, 3240
<code>\setXnoteswidthliketwocolumns@</code>	2014, 2056, 2080, 2085, 2117, 2136, 2201, 2211, 2216, 2325, 2330, 2399, 2404, 3210
<code>\showlemma</code>	54, 230, 1142, 1197, 1211
<code>\showwordrank</code>	27, 1363, 1380, 1384
<code>\sidenote@margin</code>	4329, 4449, 4467, 4523
<code>\sidenote@marginR</code>	4334, 4444, 4462
<code>\sidenotecontent@</code>	4501, 4506, 4507, 4518, 4520, 4528, 4529, 4533, 4535, 4536, 4540
<code>\sidenotemargin</code>	45, 4329
<code>\sidenotemmargin</code>	162

<code>\sidestartnumtrue</code>	16
<code>\skip</code>	1995, 1996, 2003, 2004, 2046, 2051, 2071, 2103, 2104, 2111, 2112, 2130, 2135, 2202, 2270, 2271, 2276, 2277, 2316, 2345, 2346, 2351, 2352, 2390, 2791, 2796, 2815, 2889, 2890, 2896, 2897, 2907, 2908, 2913, 2914, 2953, 2981, 2982, 2987, 2988, 3026, 3059, 3060, 3067, 3068, 3086, 3091, 3095, 3160, 3286, 3287, 3293, 3294, 3307, 3308, 3314, 3315, 3958, 3959, 3997, 3998, 4575, 4577, 4623, 4633, 4635
<code>\skip@lockoff</code>	858, 890
<code>\skipnumbering</code>	20, <u>1119</u> , 5662, 5675, 5684, 5697
<code>\spacefactor</code>	2669, 2672, 2720, 2726
<code>\spaceskip</code>	2017, 2756, 2931
<code>\splitmaxdepth</code>	1969
<code>\splitoff</code>	<u>2251</u>
<code>\splittopskip</code>	1522, 1969, 2253, 2312, 2314, 2335, 2337, 2382, 2384, 2409, 2411, 2949, 2951, 2971, 2973, 3022, 3024, 3044, 3046
<code>\spreadmath</code>	49, <u>5607</u>
<code>\spreadtext</code>	49, <u>5605</u>
<code>\stanza</code>	38, <u>4981</u>
<code>\stanza@count</code>	<u>4921</u> , 4932, 4935, 4937, 4966, 4978, 4985, 5010
<code>\stanza@chang</code>	<u>4964</u> , 4987
<code>\stanza@line</code>	<u>4964</u> , 4995, 5010
<code>\stanza@modulo</code>	4944, 4947–4949, 4958–4960, 4969, 4985
<code>\stanzaindent</code>	40, <u>4952</u>
<code>\stanzaindent*</code>	40, <u>4952</u>
<code>\stanzaindentbase</code>	38, <u>4921</u> , 4953, 4967, 4970, 4976, 5013
<code>\startlock</code>	19, <u>1110</u> , 1146, 4974
<code>\startsub</code>	19, <u>1071</u> , 1145
<code>\stepcounter</code>	3664, 4138, 4141, 4698, 5452
<code>\stepl@dc@count</code>	<u>5210</u> , 5227, 5239, 5287, 5306, 5322, 5338, 5353, 5370, 5416, 5615
<code>\StrChar</code>	3757
<code>\StrDel</code>	3736, 3743, 4252, 4253
<code>\StrGobbleLeft</code>	4759, 4764
<code>\strip@pt</code>	2122, 3079
<code>\strip@szacnt</code>	<u>4930</u>
<code>\StrPosition</code>	3750, 3751
<code>\sub@action</code>	676, <u>837</u> , 973
<code>\sub@change</code>	295, 670, 671, 677, 733, 735, 738, 740
<code>\sub@lock</code>	292, <u>570</u> , 685, 687, 689, 692, 872, 873, 875, 876, 894, 895, 897, 1607, 1679, 1681, 1682, 1684, 1767, 1803, 1805, 1807
<code>\sub@off</code>	<u>732</u> , 1081, 1082
<code>\sub@on</code>	<u>732</u> , 1075, 1076
<code>\subline@num</code>	290, 552, 553, <u>567</u> , 693, 697, 719, 753, 754, 756, 787, 830, 981, 985, 1608, 1613, 1636, 1644, 1704–1706, 4169
<code>\subline@numR</code>	709, 747, 748, 750, 780, 821, 4165
<code>\sublinenumberstyle</code>	20, <u>533</u>
<code>\sublinenumincrement</code>	18, <u>468</u>
<code>\sublinenumr@p</code>	<u>533</u> , 4165, 4169
<code>\sublinenumrep</code>	<u>533</u> , 553, 2638, 2653, 3471, 3486
<code>\sublineref</code>	42, <u>4211</u>
<code>\sublines@false</code>	293, <u>568</u> , 674, 1669
<code>\sublines@true</code>	<u>568</u> , 672, 1667

`\sublock@disp` ..... 524, 1769, 1773, 1777  
`\sublockdisp` ..... 119, 524  
`\subsection` ..... 6092, 6096, 6101, 6102  
`\subsectionmark` ..... 5710, 6093, 6097  
`\subsubsection` ..... 3652, 6109, 6110, 6113, 6114  
Sullivan, Wayne ..... 11, 12, 55, 65, 68, 136, 137, 195, 218  
`\sw@atthisline` ..... 1359, 1361, 1362, 1376, 1378, 1379  
`\sw@inthisedtext` ..... 1165, 1167, 1168, 1172, 1174, 1175, 3599, 3616  
`\sw@list@inedtext` ..... 1349, 1353, 1366, 1370, 3599, 3616  
`\sw@txt` ..... 1279,  
1281, 1288, 1289, 1292, 1300, 1302, 1303, 1305, 1307, 1309, 1310, 1312, 1357, 1359, 1374, 1376  
`\sw@txt@R` ..... 1291, 1292  
`\symlinenum` ..... 6184  
`\sza@penalty` ..... 4964, 4991, 5009

## T

`\tabelskip` ..... 5635–5637, 5643–5645  
`\tabHilfbbox` ..... 5634, 5636, 5638, 5642, 5644, 5646, 5647  
`\tabhilfbbox` ..... 5633, 5635, 5637, 5641, 5643, 5645, 5647  
`\temp@` ..... 1566, 1567, 4838, 4844, 4868  
`\textbf` ..... 1211  
`\textnormal` ..... 2413–2415  
`\textsc` ..... 1211  
`\textsuperscript` ..... 1386, 2662, 2746, 2784  
`\textwidth` ..... 4656  
`\the@sw` ..... 1305, 1312, 1335, 1337, 1350, 1354, 1363, 1367, 1371, 1380  
`\theadcolcount` ..... 5276, 5450, 5453  
`\theendpageline` ..... 4707, 4792, 4805  
`\thefootnoteA` ..... 27  
`\thelabidx` ..... 4699, 4705, 4725,  
4729, 4732, 4734, 4739, 4744, 4750, 4849, 4853, 4858, 4860, 4871, 4874, 4875, 4885, 4890, 4896  
`\theline` ..... 4142, 4144  
`\thempfn` ..... 4642, 4669  
`\thempfootnote` ..... 4642, 4669  
Theodosius ..... 11  
`\thepage` ..... 1029, 1031, 1040, 1042, 1045, 4116, 4129, 4703  
`\thepageline` ..... 4702, 4795, 4809  
`\thepstart` ..... 16, 1393, 1513, 1839, 1846, 2433  
`\thepstartL` ..... 2430  
`\thepstartR` ..... 2428  
`\thestartpageline` ..... 4707, 4791, 4802  
`\thesubline` ..... 4142  
`\thinspace` ..... 2418, 2420  
`\this@absline` ..... 1351, 1355, 1357–1359, 1368, 1372, 1374–1376  
`\this@line@list@version` ..... 638, 1006, 1013  
`\thisfootnote` ..... 2845, 2846, 2856, 2857  
`\thr@@` . 450, 875, 884, 895, 902, 1674, 1682, 2285, 2288, 2314, 2337, 2996, 2999, 3024, 3046, 4354  
`\threecolfootfmt` ..... 2267, 2295  
`\threecolfootfmtX` ..... 2978, 3007  
`\threecolfootgroup` ..... 2268, 2310



<code>\threecolfootgroupX</code>	2979, <a href="#">3021</a>
<code>\threecolfootsetup</code>	2272, <a href="#">2282</a>
<code>\threecolfootsetupX</code>	2983, <a href="#">2993</a>
<code>\threecolvfootnote</code>	2266, <a href="#">2290</a>
<code>\threecolvfootnoteX</code>	2977, <a href="#">3001</a>
<code>\tmp</code>	2611, 2612, 2619
<code>\togglefalse</code>	2045, 2129, 2790, 3085, 3383, 4302, 4312
<code>\toggletrue</code>	3273, 3277, 3356, 4291, 4308
<code>\tolerance</code>	2299, 2369, 2942, 3015
<code>\twocolfootfmt</code>	2342, <a href="#">2357</a>
<code>\twocolfootfmtX</code>	2904, <a href="#">2934</a>
<code>\twocolfootgroup</code>	2343, <a href="#">2357</a>
<code>\twocolfootgroupX</code>	2905, <a href="#">2948</a>
<code>\twocolfootsetup</code>	2347, <a href="#">2357</a>
<code>\twocolfootsetupX</code>	2909, <a href="#">2919</a>
<code>\twocolvfootnote</code>	2341, <a href="#">2357</a>
<code>\twocolvfootnoteX</code>	2903, <a href="#">2927</a>
<code>\twolines</code>	6210
<code>\twolinesbutnotmore</code>	6212
<code>\twolinesonlyinsamepage</code>	6213
<code>\txtbeforeXnotes</code>	6194

## U

<code>\unexpanded</code>	3599, 3616, 5951, 5955, 5964, 5968, 5977, 5981, 5989, 5993, 6003, 6007, 6016, 6020, 6029, 6033, 6042, 6046
<code>\unhbox</code>	1519, 2173, 2194, 2196, 2224, 2226, 2233, 2237, 3132, 3155, 3157, 3183, 3185, 5637, 5638, 5645, 5646
<code>\unkern</code>	2670
<code>\unless</code>	392, 403, 414, 1247, 1287, 1299, 1348, 1966, 2069, 3532, 3580, 3635, 3656, 3680, 3832, 3881, 3898, 3954, 3977, 3991, 4007, 4052, 4055, 4068, 4073, 4557, 4563, 4587, 4594, 4685, 4688, 5070, 5082, 6175, 6216, 6229
<code>\unpenalty</code>	2176, 2230, 3135
<code>\unvbox</code>	1526, 2021, 2065, 2091, 2157, 2171, 2190, 2220, 2262, 2761, 2810, 2833, 3117, 3130, 3151, 3179, 3947, 3980, 3994, 4012, 4627
<code>\unvxhX</code>	3109, 3123, <a href="#">3129</a>

## V

<code>\valign</code>	2254
<code>\value</code>	4948, 4960, 4965, 5449
<code>Vamana</code>	11
<code>\variab</code>	<a href="#">5258</a> , 5539, 5555, 5575, 5583, 5593, 5601
<code>\vbadness</code>	1521, 2253
<code>\vbfnoteX</code>	2846, 2857, <a href="#">2863</a>
<code>\vbox</code>	1434, 2020, 2146, 2156, 2161, 2171, 2760, 3107, 3116, 3121, 3130, 3259, 3267, 3946, 3993, 4484, 4486, 4492, 4494, 5425, 5428, 5433, 5436, 5440, 5442, 5443, 5481, 5485, 5490, 5501, 5514, 5527
<code>\vfil</code>	2254, 5426, 5429, 5434, 5437, 5440, 5443
<code>\vl@dbfnote</code>	<a href="#">2698</a>
<code>\vl@dcnote</code>	4432, 4436, <a href="#">4442</a>
<code>\vl@dlsnote</code>	4402, 4406, <a href="#">4442</a>

\vl@drsnote .....	4417, 4421, <a href="#">4442</a>
\vnumfootnoteX .....	<a href="#">2867</a> , 2881
\vrule .....	5425, 5428, 5433, 5436, 5440
\vsize .....	3575, 3647
\vsplit .....	1525, 2261

## W

\wd .....	1513, 1553, 2150, 2165, 3111, 3125, 5151, 5216, 5217, 5494, 5503, 5506, 5516, 5519, 5528, 5635, 5636, 5643, 5644
Whitney, Ron .....	11
\widowpenalty .....	1494, 1880
\widthliketwocolumnstrue .....	45
\WithSuffix .....	4956, 5999, 6012, 6025, 6038
\wrap@edcrossref .....	<a href="#">4200</a> , 4205, 4208, 4211, 4214
Wujastyk, Dominik .....	10

## X

\X@doreinfeet .....	3968, 3971, 3972, <a href="#">3976</a>
\x@lemma .....	1199–1201
\Xafterlemmaseparator .....	33, 6191
\Xafternote .....	35, 6225
\Xafternumber .....	31, 6186
\Xafterrule .....	36, 6195
\Xaftersymlinenum .....	31, 6188
\Xarrangement .....	29, <a href="#">1973</a> , 6176–6179
\Xarrangement@normal .....	<a href="#">1985</a> , 3633
\Xarrangement@paragraph .....	<a href="#">2093</a>
\Xarrangement@threecol .....	<a href="#">2264</a>
\Xarrangement@twocol .....	<a href="#">2339</a>
\XarrangementX .....	6173
\Xbeforelemmaseparator .....	33, 6192
\Xbeforenotes .....	36, 6205
\Xbeforenumber .....	31, 6185
\Xbeforesymlinenum .....	31, 6187
\Xbhooknote .....	34, 6182
\Xboxendlinenum .....	6209
\Xboxendlinenum@appref .....	4273
\Xboxlinenum .....	32, 6206
\Xboxlinenumalign .....	6207
\Xboxlinenumalign .....	32
\Xboxstartlinenum .....	6208
\Xboxstartlinenum@appref .....	4272
\Xboxsymlinenum .....	32, 6183
\Xdo@feet .....	3937, 3940, 3941, <a href="#">3945</a>
\xedindex .....	<a href="#">5115</a> , 5135, 5146
\xedlabel .....	<a href="#">5113</a> , 5152
\xedtext .....	<a href="#">5111</a> , 5145
\Xendafterlemmaseparator .....	33, 6239
\Xendafternote .....	37
\Xendbeforelemmaseparator .....	33, 6238

\Xendbhooknote .....	34, 6232
\Xendboxendlinenum .....	6236
\Xendboxendlinenum@apprefwithpage .....	4276
\Xendboxendlinenumalign .....	32
\Xendboxlinenum .....	32, 6233
\Xendboxlinenumalign .....	32, 6234
\Xendboxstartlinenum .....	6235
\Xendboxstartlinenum@apprefwithpage .....	4275
\Xendboxstartlinenumalign .....	32
\Xendinplaceoflemmaseparator .....	33, 6240
\Xendinplaceofnumber .....	31
\Xendinsertsep@false .....	3504, 3517
\Xendinsertsep@true .....	3351
\Xendlemmadisablefontselection .....	34
\Xendlemmaseparator .....	33, 6237
\Xendmorethantwolines .....	30, 6231
\Xendmorethantwolines@apprefwithpage .....	4267
\Xendmorethantwolinesapprefwithpage .....	45
\Xendnonumber .....	31
\Xendnotefontsize .....	34
\Xendnotenumfont .....	33
\Xendparagraph .....	37
\Xendsep .....	37
\Xendtwolines .....	30, 6230
\Xendtwolines@apprefwithpage .....	4266
\Xendtwolinesapprefwithpage .....	45
\Xendtwolinesbutnotmore .....	30
\Xendtwolinesbutnotmoreapprefwithpage .....	45
\Xendtwolinesonlyinsamepageapprefwithpage .....	45
\XendXafterlemmaseparator .....	6239
\XendXbeforelemmaseparator .....	6238
\XendXinplaceoflemmaseparator .....	6240
\XendXlemmaseparator .....	6237
\XendXmorethantwolines .....	6231
\XendXtwolines .....	6230
\Xhangindent .....	34
\Xhsizethreecol .....	35, 6181
\Xhsizetwocol .....	35, 6180
\xifinlist .....	1024, 1025, 1035, 1036, 1541, 1567, 3527, 6133, 6136, 6138, 6145, 6146
\xindy@true .....	51
\xindyhyperref@true .....	58
\Xinplaceoflemmaseparator .....	33, 6193
\Xinplaceofnumber .....	31, 6189
\Xinsertparafootsep .....	2180, 2239, 3187
\Xledsetnormalparstuff .....	2032, 2181, 3192
\xleft@appenditem .....	370
\Xlemmadisablefontselection .....	34
\Xlemmaseparator .....	32, 3930, 6190
\xlineref .....	43, 4208, 4297, 4301, 4311, 4705
\Xmaxhnotes .....	37, 6204

\Xmorethantwolines	30, 6211
\Xmorethantwolines@appref	4262
\Xmorethantwolinesappref	45
\Xnolemmaseparator	33, 3930
\Xnonbreakableafternumber	31, 6203
\Xnonumber	30, 6202
\Xnonumberinfootnote	6198, 6202
\Xnotefontsize	33
\Xnotenumfont	33
\Xnoteswidthliketwocolumns	37
\Xnumberonlyfirstinline	29, 6196
\Xnumberonlyfirstintwolines	30, 6197
\Xonlypstart	31, 6201
\xpageref	43, <a href="#">4205</a> , 4301, 4311
\Xparafootsep	35, 6220
\Xparindent	34
\Xpstart	31, 393, 404, 415, 6199
\Xpstarteverytime	31, 6200
\xpstartref	43, <a href="#">4214</a>
\Xragged	36
\xright@appenditem	<a href="#">364</a> , 810, 811, 813, 814, 819, 826, 828, 835, 840, 842, 844, 847, 849, 851, 869, 871, 880, 891, 893, 900, 919, 920, 922, 923, 929, 930, 932, 933, 946, 947, 949, 950, 966, 983, 997, 1335, 1337, 1916, 1920, 1928, 1930, 1934, 1936, 1943, 1946, 1949, 1954, 1957, 1960, 2703, 2707, 2846, 2857, 3596, 3613, 4164, 4168, 4402, 4406, 4417, 4421, 4432, 4436
\xspaceskip	2017, 2756, 2931
\xsublineref	43, <a href="#">4211</a> , 4301, 4311
\Xsymlinenum	30, 6184
\Xtwolines	30, 6210
\Xtwolines@appref	4261
\Xtwolinesappref	45
\Xtwolinesbutnotmore	6212
\Xtwolinesbutnotmoreappref	45
\Xtwolinesonlyinsamepage	30, 45, 6213
\Xtxtbeforenotes	36, 6194
\Xunvxh	2148, 2163, <a href="#">2170</a>
\xxref	44, <a href="#">4239</a>

## Z

\z@skip	1970, 2017, 2756, 2931
\zz@@@	<a href="#">4098</a> , 4108, 4121, 4242, 4247

## Change History

v0.1.0.	
General: First public release	1
v0.2.0.	
General: Added tabmac code, and extended indexing	1
\ifl@dmemoir: Added \ifl@dmemoir for memoir class having been used	59
\morenoexpands: Added \l@dtabnoexpands to \no@expands	99
\reledmac@error: Added \eledmac@error and replaced error messages	60
v0.2.1.	
\@lab: Removed page setting from \@lab	197
General: Added text about normal labeling	44
Bug fixes and match with mempatch v1.8	1
Major changes to insert code when memoir is loaded	193
\dextrafeet: Renamed \dextrafeet to \l@ddextrafeet	190
\edlabel: Tweaked \edlabel to get correct page numbers	195
\l@ddodoreinxtrafeet: Renamed \dodoreinxtrafeet to \l@ddodoreinxtrafeet	191
\morenoexpands: Removed some \lets from \no@expands. These were in edmac but Peter Wilson feels that they should not have been as they disabled page/line refs in a footnotes	99
\zz@@@: Minor change to \zz@@@	195
v0.2.2.	
General: Improved paragraph footnotes	1
New Dekker example	1
Used \providecommand for \@gobblethree to avoid clash with the amsfonts pack- age	64
\footfudgefiddle: Added \footfudgefiddle	135
\line@list@stuff: Added initial write of page number in \line@list@stuff	92
\para@footsetup: Added \footfudgefiddle to \para@footsetup	135
\para@footsetupX: Added \footfudgefiddle to \para@footsetupX	164
v0.3.0.	
\@lab: Replaced \the\line@num by \linenumr@p\line@num in \@lab, and similar for sub-lines	197
\@nl@reg: Added a bunch of code to \@nl for handling \setlinenum	83
General: Includes edstanza and more	1
\ledlinenum: Added \linenumr@p and \sublinenumr@p to \leftlinenum and \rightlinenum	75
\linenumberlist: Added \linenumberlist mechanism	65
\printendlines: Added \linenumr@p and \sublinenumr@p to \printendlines	175
\printlines: Added \linenumr@p and \sublinenumr@p to \printlines	152
\sublinenumr@p: Added \linenumberstyle and \sublinenumberstyle	74
v0.3.1.	
General: Not released. Added remarks about the parallel package	1
v0.4.0.	
\@iiiminipage: Modified kernel \@iiiminipage and \endminipage to cater for criti- cal footnotes	210
General: Added final/draft options	58
Added minipage, etc., support	1
ledgroup: Added ledgroup environment	211
ledgroupsize: Added ledgroupsize environment	211

\edtext: Added \showlemma to \edtext	99
\l@dfeetendmini: Added \l@dfeetbeginmini, \l@dfeetendmini and all their supporting code	208
\mpnormalfootgroup: Added \mpnormalfootgroup	133
\mpnormalvfootnote: Added \mpnormalvfootnote	132
\showlemma: Added \showlemma	64
\Xarrangement@normal: Added minpage footnote setup to \footnormal	131
v0.4.1.	
General: Added code for changing \@docclearpage	193
Not released. Minor editorial improvements and code tweaks	1
Only change \@footnotetext and \@footnotemark if memoir not used	154
\edindex: Let eledmac take advantage of memoir's indexing	215
\print@Xnotes: Added \@opXfeet	191
\Xdo@feet: Changed \Xdo@feet code for easier extensions	191
v0.5.0.	
\@footnotetext: Enabled regular \footnote in numbered text	154
\@xympar: Eliminated \marginpar disturbance	202
General: Added left and right side notes	203
Added sidenotes, familiar footnotes in numbered text	1
v0.5.1.	
General: Added moveable side note	203
Fixed right line numbers killed in v0.5	1
\affixline@num: Changed \affixline@num to cater for sidenotes	120
ledgroupsize: Only change \hsize in ledgroupsize environment otherwise page number can be in wrong place	211
\l@dgetsidenote@margin: Added \sidenotemargin and \sidenote@margin	203
v0.6.0.	
\@lopR: Added \@pend, \@pendR, \@lopL and \@lopR in anticipation of parallel processing	84
\@nl@reg: Added \fix@page to \@nl	83
Extended \@nl to include the page number	83
General: Fixed long paragraphs looping	1
Fixed minor typos	1
Prepared for eledpar package	1
\fix@page: Added \last@page@num and \fix@page	84
\new@line: Extended \new@line to output page numbers	93
\vl@dbfnote: Changed \l@dbfnote and \vl@dbfnote as originals could give incorrect markers in the footnotes	154
v0.7.0.	
\@nl@reg: Added \@nl@reg	83
\@ref@reg: Added \@ref@reg	90
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	60
Replaced all \interfootnotelinepenalty, etc., by just \interfootnotelinepenalty	1
Tidying up for eledpar and ledarab packages	1
\affixline@num: Added skipnumering to \affixline@num	120
\do@actions@fixedcode: Added \do@actions@fixedcode	119

\do@actions@next: Added number skipping to \do@actions	118
\do@insidelinehook: Added \do@linehook for use in \do@line	116
\endnumbering: Changed \endnumbering for eledpar	67
\f@x@l@cks: Added \ch@cksub@l@ck, \ch@ck@l@ck and \f@x@l@cks	122
\footsplitskips: Added \footsplitskips for use in many footnote styles	130
\get@linelistfile: Added \get@linelistfile	81
\ifledRcol@: Added \l@dunpstartsL, \ifl@dpairing and \ifpst@rted for/from eledpar	65
\initnumbering@reg: Added \initnumbering@reg	66
\l@advance@parledgroup@beforenormalnotes: Added \l@dunboxmpfoot containing some common code	210
\l@dcsnotetext@r: Added \l@demptyd@ta	116
\l@dgetline@margin: Added \l@dgetline@margin	72
\l@dgetlock@disp: Added \l@dgetlock@disp	74
\l@dgetsidenote@margin: Added \l@dgetsidenote@margin	203
\l@drsn@te: Added \l@dlsn@te and \l@drsn@te for use in \do@line	117
\l@dzeropenalties: Added \l@dzeropenalties	112
\ledlinenum: Added \ledlinenum for use by \leftlinenum and \rightlinenum	75
\line@list@stuff: Deleted \page@start from \line@list@stuff	92
\list@clearing@reg: Added \list@clearing@reg	81
\n@num: Added \n@num	89
\normalbfnoteX: Removed extraneous space from \normalbfnoteX	158
\resumenumbering: Changed \resumenumbering for eledpar	68
\setprintendlines: Added \setprintendlines for use by \printendlines	174
\setprintlines: Added \setprintlines for use by \printlines	150
\skipnumbering: Added \skipnumbering and supports	96
\sublinenumincrement: Added \firstlinenum, \linenumincrement, \firstsublinenum and \linenumincrement	73
\sublinenumr@p: Using \linenumrep instead of \linenumr@p	74
Using \sublinenumrep instead of \sublinenumr@p	74
\vnumfootnoteX: Removed extraneous space from \vnumfootnoteX	159
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 us 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.	16
Possibility to number the pstart with the commands \numberpstarttrue.	1

\ifledRcol@: Added \ifledRcol and \ifnumberingR for/from eledpar . . . . .	65
v0.12.1.	
General: Don't number \pstarts of stanza. . . . .	1
The numbering of \pstarts restarts on each \beginnumbering. . . . .	1
v0.13.0.	
General: New stanzaindentsrepetition counter to repeat stanza indents every $n$ verses. . . . .	39
New stanzaindentsrepetition counter: to repeat stanza indents every $n$ verses. . . . .	1
\managestanza@modulo: New stanzaindentsrepetition counter to repeat stanza indents every $n$ verses. . . . .	220
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. . . . .	195
v0.15.0.	
General: Line numbering can be reset at each pstart. . . . .	70
Possibility to print \pstart number inside. . . . .	16
\affixline@num: Line numbering can be disabled. . . . .	120
\ifinserthangingsymbol: New management of hangingsymbol insertion, preventing undesirable insertions. . . . .	218
\printlines: Line numbering can be reset at each pstart. . . . .	152
v0.17.0.	
\ifinserthangingsymbol: New new management of hangingsymbol insertion, preventing undesirable insertions. . . . .	218
v1.0.0.	
General: \lemma can contain commands. . . . .	23
Debug in lineation command . . . . .	18
New generic commands to customize footnote display. . . . .	28, 184
Options nonum and noseup in \Xfootnote. . . . .	22
Options of \Xfootnotes. . . . .	128
Possibility to have commands in sidenotes. . . . .	45
Some compatibility break with eledmac. Change of name: eledmac. . . . .	1
\morenoexpands: Change to be compatible with new features . . . . .	99
v1.0.1.	
General: Correction on \Xnumberonlyfirstinline with lineation by pstart or by page. . . . .	29
v1.1.0.	
General: Add \labelpstarttrue. . . . .	16
Add \Xnumberonlyfirstintwolines . . . . .	30
Add \Xpstart and \Xonlypstart . . . . .	31
New hook to add arbitrary code at the beginning of the notes . . . . .	34
New options for block of notes. . . . .	36
New package option: parapparatus. . . . .	1
New tools to change order of series . . . . .	183
Sectioning commands. . . . .	51
\preXnotes: New skip \preXnotes@ . . . . .	170
\settoggle@series: \settoggle@series switch the global value of the toggle, not only the local value. . . . .	185



v1.2.0.	
\endquote: Compatibility of \ledchapter with the <i>memoir</i> class. . . . .	242
\preXnotes: Debug in familiar footnotes (bug introduced by v1.1). . . . .	170
v1.3.0.	
\endquote: <i>Quotation</i> and quote environment inside numbered sections. . . . .	242
v1.4.0.	
General: Compatibility with LuaTeX of RTL notes. . . . .	1
\edtext: Compatibility of \edtext with the right-to-left direction (with Polyglossia). . .	99
\ledsetnormalparstuffX: Direction of footnotes with polyglossia. . . . .	167
\newseries@: Remembers the language of the lemma, in order to create a correct direc- tion for the footnote separator. . . . .	179
\rbracket: Switch the right bracket to a left bracket when the lemma is RTL (needs polyglossia or LuaTeX). . . . .	145
v1.4.1.	
\affixside@note: Remove spurious spaces. . . . .	207
\endquote: New option <i>noquotation</i> . . . . .	242
\labelrefsparsesubline: Fix bug with \edlabel. . . . .	196
\vl@dbfnote: Compatibility of standard footnotes with eledmac when these footnotes contain any commands. . . . . .	154
v1.4.2.	
General: Debug with some special classes. . . . .	1
v1.4.3.	
General: Add \Xnonbreakableafternumber. . . . .	31
Spurious space after familiar footnotes. . . . .	1
v1.4.4.	
General: Label inside familiar footnotes. . . . .	1
v1.4.5.	
General: Bug with komasscript + eledpar + chapter. . . . .	1
v1.4.6.	
General: Bug with memoir class introduced by 1.4.5. . . . .	1
v1.4.7.	
\endquote: Compatibility of sectioning commands with \autopar. . . . .	242
v1.4.8.	
General: Corrects a bug with parallel texts introduced by 1.1. . . . .	1
v1.4.9.	
\normalbfnoteX: Allow to redefine \thefootnoteX with alph when some packages are loaded. . . . .	158
v1.5.0.	
General: Correct indexing when the call is made in critical notes. . . . .	212
\do@insidelinehook: Added \do@insidelinehook for use in \do@line . . . . .	116
\edindex: Compatibility with imakeidx package, and possibility to use multiple index with \edindex. . . . .	215
v1.5.1.	
\managestanza@modulo: Correct stanzaindentsrepetition counter . . . . .	220
\normalvfootnoteX: Fix bug with normal familiar footnotes when mixing RTL and LTR text. . . . .	156
v1.6.0.	
\newverse: Add \falseverse macro. . . . .	221
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. . . . .	110
\ifinserthangingsymbol: Hang verse is now not automatically flush right. . . . .	218
\l@dunhbox@line: Move the call to \inserthangingsymbol to allow use \hfill in-side. . . . .	114
\pend: Spurious space in \pend. . . . .	111
v1.7.0.	
General: New features for managing page breaks. . . . .	53
v1.8.0.	
General: Compatibility with parledgroup option of eledpar package. . . . .	1
If imakeidx and hyperref are loaded, adds hyperref in the index. . . . .	212
\endquote: Correction of sectioning commands in parallel texts. . . . .	242
\get@index@command: Debug \get@index@command and compatibility with hyperref package. . . . .	214
\newhookcommand@series@reload: Debug \beforenotesX and \maxhnotesX which did not work. . . . .	187
\prevpage@num: Correct \parafootsep when using with ledgroup. . . . .	140
v1.8.1.	
General: Debug endnotes when more than one series is used (change the position where tools for endnotes are defined). . . . .	171
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. . . . .	212
v1.9.0.	
\doxtrafeet: Add \fnpos to choice the order of footnotes. . . . .	190
\l@dfeetendmini: Add \mpfnpos to choice the order of footnotes in minipage / led-group. . . . .	208
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
v1.10.1.	
General: Compatibility with cleveref. . . . .	1
v1.10.2.	
General: Compatibility of stanza with v1.8a of babel-greek. . . . .	1
v1.10.3.	
General: Debug of cross-referencing. . . . .	1
v1.10.4.	
General: Debug of critical notes in edtabular environment. . . . .	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. . . . .	34
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. . . . .	82
General: Add \ledinnernote and \ledouternote commands. . . . .	45
Add \Xendparagraph and related settings. . . . .	37
Add hyperlink to crossref (needs hyperref package). . . . .	42
Compatibility with musixtex. . . . .	1
Debug eledmac sectioning command after using \resumenumbering. . . . .	1
Ensure that imakeidx is loaded <i>before</i> eledmac . . . . .	212
New hooks: \Xafterrule and \afterruleX . . . . .	36
New options for ragged-paragraph notes . . . . .	36
New sectioning commands. . . . .	51
Optional arguments for \pstart and \pend. . . . .	16
\AtEveryPstart: New optional argument for \pstart, to execute code before it. . .	110
\edindex: Use correctly default index when imakeidx is loaded. . . . .	215
\endquote: \ledxxx sectioning commands are deprecated and replaced by \eledxxx commands. . . . .	242
\ifledRcol@: Add \ifledRcol@ for eledpar . . . . .	65
\initnumbering@reg: \beginnumbering is defined only on eledmac, not on eledpar. .	66
\l@dcsnote: \l@dlsnote, \l@drsnote and \l@dcsnote defined only one time, in eledmac, including needs for eledpar case. . . . .	204
\l@dgetsidenote@margin: \sidenotemargin is now directly defined in eledmac to be able to manage eledpar. . . . .	203
\l@dunhbox@line: \do@line is split in more little commands. . . . .	114
\newhookcommand@series@reload: Debug \beforenotesX and \maxhnotesX which did not work when called after \footparagraphX. . . . .	187
Debug \Xbeforenotes and \Xmaxhnotes which did not work when called after \footparagraph. . . . .	187
\pend: New optional argument for \pend, to execute code after it. . . . .	111
\stanza: &can have an optional argument: content to be printed after. . . . .	221
\Stanza can have an optional argument: content to be printed before. . . . .	221
Add \newverse macro, \falseverse deprecated. . . . .	221

## v1.12.1.

\wrap@edcrossref: Fix spurious spaces. . . . .	198
--	-----

## v1.12.2.

\l@dunhbox@line: Fix a bug with critical notes at the tops of pages (added by v12.0.0)	114
--	-----

## v1.12.3.

General: Add macros for new messages since v0.7 . . . . .	60
Correct bug with side and familiar notes in tabular environments. . . . .	1
Debug \eledxxx with some paper size . . . . .	1
Debug \ledinnernote and \ledouternote commands in the top of pages. . . . .	45
Debug left and right notes (bugs added by 1.12.0) . . . . .	1
Underline lemma in \eledxxx when using draft mode. . . . .	1
\flag@end: \flag@start and \flag@end are now defined only one time for eledmac and eledpar . . . . .	93
\flag@start send a error message when a \edtext is done without insert (note) .	93
\reledmac@error: Replaced error messages . . . . .	60

## v1.12.4.

General: Debug spurious page breaks before \chapter (bug added in 1.12.0) . . . . .	1
---	---

v1.12.5.	
\@edindex@hyperref: Debug \edindex when hyperref is not loaded	216
\@ssect: Debug \eledchapter in parallel with memoir	245
\doinsidelinehook: Added \dolinehook and \doinsidelinehook	116
\endnumbering: Allow to mix parallel columns and normal text when using \pausenumbering	67
\l@dgobblearg: \l@dgobblearg becomes \l@gobbeloptarg	228
\l@drestoreforedtext: Debug optional arguments of \Xfootnote in tabular context	228
\resumenummering: Debug \resumenummering	68
v1.12.7.	
\wrap@edcrossref: \wrap@edcrossref is now robust	198
v1.12.8.	
\flag@end: \flag@start do not send a error message when a \edtext is done without insert (note) but have a endnote	93
v1.13.0.	
General: Add \Xnoteswidthliketwocolumns and \noteswidthliketwocolumnsX	37
Added widthliketwocolumns option	58
\newhooktoggle@series@reload: Add \newhookcommand@toggle@reload	187
\para@footsetupX: In \para@footsetupX, use \columnwidth instead of \hsize	164
\settoggle@series: \settoggle@series can take an optional arguments to reload series setup.	185
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 \pstart preceded by a sectioning command.	1
\thepstart: Add \l@dzeropenalties in \pstart	110
v1.13.2.	
General: Fix bug with normal footnotes, added by v1.13.0.	1
\ifledRcol@: Add \ifl@dpadding for eldpar	65
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.	171
Delete previous override of \l@d@@wrindexhyp at the beginning of a document when hyperref is not loaded.	218
Move gobbling command	64
Provide \@gobblefour	64
\edindex: Let eldmac take advantage of imakeidx even when memoir class is used	215
v1.14.1.	
\@ssect: Debug sectioning commands when using both handout and hyperref package.	248
v1.14.2.	
\@ssect: Debug \edtext after starred sectioning commands when using memoir class.	245
v1.15.0.	
\@edtext@level: New boolean \if@edtext@.	99
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.	16
New tools to prevent ambiguous references in lemma	24

\arrangementX@threecol: Correct bug with paragraphed familiar footnotes setting.	163
\endsub: Restore subline feature (disabled by mistake in v1.8.0).	94
\if@lemmacommand@: New boolean \iflemmacommand@.	103
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.	92
v1.16.0.	
General: \edtext is now defined only in eledmac, not in eledpar. Debug wrong numbering when using \sameword + eledpar + \tag command.	99
Compatibility of standard footnotes with some biblatex styles.	1
New \stanzaindent command.	1
v1.16.1.	
\edlineref: \lineref is not defined if defined by some other package, like lineno. Eledmac provides \edlineref instead.	199
v1.17.0.	
\edtext: Error message when calling \edtext outside of a numbered paragraph.	99
v1.18.0.	
\@edindex@hyperref: Fix spurious space with \edindex when using imakeidx/indextools + hyperref.	216
General: Add \Xpstarteverytime	31
Compatibility with Lua $\TeX$ RTL languages.	1
Debug \Xonlypstart when using \Xnumberonlyfirstinline and the current line number differs from the previous.	31
\edlabel: \edlabel is now defined only one time for both eledmac and eledpar	195
\ifledRcol@: Add \ifl@dprintingpages and \@dprintingcolumns for eledpar	65
\l@d@section: Option parapparatus works for endnotes.	172
\print@line: Compatibility with Lua $\TeX$ RTL languages.	115
\printlinefootnote: Code refactoring in \printlinefootnote: the printing of the numbers are factorized in \printlinefootnotearea	146
\printpstart: Debug \Xpstart with parallel pages and columns (eledpar)	146
v1.19.0.	
General: \Xmaxhnotes 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.	130
\xxref: \xxref works also with right side numbers, when \@Rlineflag is not empty.	200
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 \Xendboxlinenum	32
Add \Xtwolines and \Xmorethantwolines hooks	30
Add series option.	1
Correct \Xinplaceofnumber 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 Lua $\TeX$ .	1
Fix issues with RTL text in notes when using Lua $\TeX$ .	1
Options fulllines in \Xfootnote.	22

The <code>\newifs</code> are not followed by boolean values set to false, because it is the $\TeX$ default setting. . . . .	1
<code>\printlines</code> : Added <code>\ifl@d@Xmorethantwoline</code> s and <code>\ifl@d@Xmorethantwoline</code> to <code>\printlines</code> . . . . .	152
<code>\stanza</code> : <code>&amp;</code> and <code>\&amp;</code> can be preceded by spaces. . . . .	221
<code>\xxref</code> : Debug <code>\xxref</code> when not loading <code>eledpar</code> (fix bug added in 1.19.0). . . . .	200
v1.21.0.	
<code>\@edindex@hyperref</code> : Look at the <code>hyperindex</code> option of <code>hyperref</code> before inserting <code>hyperref</code> . . . . .	216
General: <code>\AtEveryPstart</code> and <code>\AtEveryPend</code> are now compatible with <code>\autopar</code> . . .	1
<code>\Xafterrule</code> and <code>\afterruleX</code> features no longer create problems of overflowing at the bottom of the page. . . . .	1
<code>\chapter</code> inside optional argument of <code>\pstart</code> works when typesetting parallel pages . . . . .	1
<code>\preXnotes</code> and <code>\prenotesX</code> features no longer create problems of overflowing at the bottom of the page. . . . .	1
<code>\seriesatbegin</code> and <code>\seriesatbegin</code> more efficient . . . . .	183
Add <code>\applabel</code> and related . . . . .	44
Add <code>\beforenotesX</code> and <code>\Xbeforenotes</code> features for notes set in two and three column. . . . .	1
Add <code>\hidenumbers</code> . . . . .	20
Add <code>\Xcolalign</code> and <code>\colalignX</code> hooks . . . . .	35
Add <code>\Xendtwoline</code> s, <code>\Xendmorethantwoline</code> s, <code>\Xendtwoline</code> sbutnotmore and <code>\Xendtwoline</code> sonlyinsamepage. . . . .	30
Add <code>\Xparindent</code> and <code>\hangindentX</code> . . . . .	34
Add <code>\Xtwoline</code> sbutnotmore and <code>\Xtwoline</code> sonlyinsamepage. . . . .	1
Add <code>nocritical</code> , <code>noend</code> , <code>nofamiliar</code> and <code>noledgroup</code> options. . . . .	1
Add <code>noeledsec</code> package option . . . . .	1
Debug <code>\beforenotesX</code> <code>\maxhnotesX</code> <code>\noteswidthliketwocolumnsX</code> and <code>\afterruleX</code> with footnotes set in two and three columns. . . . .	1
Fix bug when a <code>\Xfootnote</code> follows a <code>\Xendnote</code> in the second argument of <code>\edtext</code> (bug added in <code>eledmac</code> 1.0.0) . . . . .	1
Fix bug with <code>\maxhnotesX</code> when using <code>\foottwocolX</code> or <code>\footthreecolX</code> . . . . .	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 <code>\doendnotes</code> . <code>oldprintnnumspace</code> option allows to come back to previous setting . . . . .	1
<code>parapparatus</code> option works now with familiar footnotes. . . . .	1
Provide <code>\@gobblefive</code> . . . . .	64
<code>\l@d@section</code> : <code>\endnotes</code> take five arguments. . . . .	172
<code>\ledinnotemark</code> : Add <code>\ledinnotemark</code> . . . . .	214
<code>\ledsetnormalparstuffX</code> : <code>\ledsetnormalparstuff</code> is deprecated and becomes <code>\ledsetnormalparstuffX</code> and <code>\Xledsetnormalparstuff</code> . . . . .	167
<code>\n@num</code> : <code>\n@num@ref</code> deleted . . . . .	89
<code>\n@num</code> defined only one time for both <code>Eledmac</code> and <code>Eledpar</code> . . . . .	89
<code>\newhookcommand@series</code> : <code>\newhookcommand@series</code> can take an optional argument. . . . .	186
<code>\newhooktoggle@series</code> : <code>\newhooktoggle@series</code> can take an optional argument. . . . .	187
<code>\print@footnoteXrule</code> : Code refactoring: the spaces after the footnote rules are directly managed in <code>\print@Xfootnoterule</code> and <code>\print@footnoteXrule</code> . . . . .	169
<code>\seriesatend</code> : Fix spurious space in <code>\seriesatend</code> . . . . .	183

\skipnumbering: \skipnumbering defined only one time for both Eledmac and Eled-	
par. . . . .	96
Correct \skipnumbering for stanza. . . . .	96
Delete \skipnumbering@reg. . . . .	96
v1.22.0.	
General: Add \doendnotesbysection command. . . . .	22
Add option for lemma separator inside endnotes . . . . .	33
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. . . . .	22
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 hyper-	
links. . . . .	214
v1.22.1.	
General: Fix bug (added on v1.22.0) with \Xinplaceofnumber hook. . . . .	1
\prevpage@num: Correct double symbol when using both \parafootsep and \Xsymlinenum.	
. . . . .	140
v1.23.0.	
\@edtext@level: The boolean \if@edtext@ becomes the counter \edtext@level. . . . .	99
General: Add \Xboxlinenumalign and \Xendboxlinenumalign. . . . .	32
Add \Xboxstartlinenum, \Xendboxstartlinenum, \Xboxendlinenum, \Xendboxendlinenum.	
. . . . .	32
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. . . . .	24
Fix bug with \sameword when the same lemma is used for multiple notes or for nested	
\edtexts. . . . .	24
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 upgrad-	
ing the eledmac version. . . . .	1
Optional argument of \sameword can be a comma separated list of \edtext depth. . . . .	24
\lemma: Fix spurious space after \lemma command . . . . .	103
\newseries@: Prevent spurious spaces when \Afootnote and similar commands are	
followed by spaces (bug added on 1.0.0). . . . .	179
\sameword: In order to allow use of \sameword with inputenc, we detokenize its manda-	
tory argument before using it in control sequence names. . . . .	106
v1.23.1.	
General: Fix bug with \lemma command in the right side. . . . .	1
v1.23.2.	
General: Compatibility with L <sup>A</sup> T <sub>E</sub> X's release 2015. . . . .	1
v1.24.0.	
General: We can reinitialize \AtEveryPstart and \AtEveryPend providing to it an	
empty argument. . . . .	1
v1.24.1.	
General: \lemma is disabled when using 'nocritical' option. . . . .	1

v1.24.2.	
General: Fix incompatibility between ‘nofamiliar’ option and ‘memoir’ package. . . . .	1
v1.24.3.	
General: Restore marginal numbers and notes with sectioning command (bug introduced in v1.21.0) . . . . .	1
v1.24.4.	
General: Fix spurious space with \edindex when using xindy+hyperref option. . . . .	1
v1.24.5.	
General: Fix bug of indent, when a added in 1.1.0, when a \beginnumbering immediately follow a sectioning command. . . . .	1
v2.0.0.	
\@iiiminipage: Patch \@iiiminipage instead of redefining it. . . . .	210
\@xympar: Patching \@xympar instead of redefining it . . . . .	202
General: \@makecol, \@reinserts and \@doclearpage are patched instead of begin redefined . . . . .	193
\doxtrafeeti becomes \do@feetX; \doxtrafeetii becomes \Xdo@feet; \@opxtrafeeti becomes \@opfeetX; \doreinxtrafeetii becomes \X@doreinfeet; \doreinxtrafeeti becomes \@doreinfeetX. . . . .	193
Add \Xendinplaceofnumber hook. . . . .	1
Add \Xendnonumber hook. . . . .	1
Add nonum option for endnotes. . . . .	1
Fix bug when printing only one series of endnotes, but wanted to keep endnotes for other series. . . . .	1
In order to have a more consistent name’s convention, many names has been changed. . . . .	1
Many L <sup>A</sup> T <sub>E</sub> X’s output macros are now patched and not override. . . . .	1
Package’s name becomes reledmac. . . . .	1
Patch \@footnotemark instead of redefine it . . . . .	154
Suppress indexing command specific to memoir. . . . .	215
\endminipage: Patch \endminipage instead of redefining it. . . . .	210
\initnumbering@quote: \initnumbering@sectcmd becomes \initnumbering@quote . . . . .	242
\l@advance@parledgroup@beforenormalnotes: Some conde of \l@dumboxmpfoot moved to \l@advance@parledeggroup@beforenormalnotes . . . . .	210
\newseries@: One endnotes file by series. . . . .	181