

# 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 tools in order to typeset scholarly editions. It is based on the **eledmac** package, which was based on the **ledmac** package, which was based on the **edmac** T<sub>E</sub>X package.

It can be used in combination with **reledpar** in order to typeset two texts in parallel, like an original text and its translation in a 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 every possible case). Examples starting with “1-” are for basic uses, those starting with “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 at:  
<http://geekographie.maieul.net/146>

## Contents

<b>1 Introduction</b>	<b>11</b>
1.1 Aim of the package . . . . .	11
1.2 History . . . . .	12
1.2.1 edmac . . . . .	12
1.2.2 ledmac . . . . .	13
1.2.3 eledmac . . . . .	14

---

\*This file (**reledmac.dtx**) has version number v2.21.0, last revised 2017/06/04.

<sup>†</sup>maieul at maieul dot net

1.2.4 <code>reledmac</code> . . . . .	14
1.3 Bibliography . . . . .	14
<b>2 How the package works — the problem of the number of <math>\LaTeX</math> runs</b>	<b>14</b>
<b>3 Compatibility warning</b>	<b>15</b>
<b>4 Options</b>	<b>15</b>
4.1 Specific features . . . . .	15
4.2 Optimizing package performance . . . . .	16
<b>5 Text lines and paragraphs numbering</b>	<b>16</b>
5.1 Text lines numbering . . . . .	16
5.2 Paragraphs . . . . .	17
5.2.1 Basics . . . . .	17
5.2.2 Automatically producing <code>\pstart ... \pend</code> . . . . .	18
5.2.3 Content before specific <code>\pstart</code> and after specific <code>\pend</code> . . . . .	18
5.2.4 Content before every <code>\pstart</code> and after every <code>\pend</code> . . . . .	19
5.2.5 Numbering paragraphs ( <code>\pstart</code> ) . . . . .	19
5.2.6 Languages written in Right to Left . . . . .	20
5.2.7 Memory limits . . . . .	20
5.3 Lineation commands . . . . .	20
5.3.1 Disabling lineation . . . . .	20
5.3.2 Setting lineation start and step . . . . .	21
5.3.3 Setting lineation reset . . . . .	21
5.3.4 Setting line number margin . . . . .	21
5.3.5 Other settings . . . . .	22
5.4 Changing the line numbers . . . . .	22
5.4.1 Sublineation . . . . .	22
5.4.2 Locking lineation . . . . .	22
5.4.3 Setting and changing line number . . . . .	22
5.4.4 Line number style . . . . .	23
5.4.5 Skipping and hiding number . . . . .	23
5.5 Executing code at each line . . . . .	24
<b>6 Apparatus commands</b>	<b>24</b>
6.1 Terminology . . . . .	24
6.2 Critical notes . . . . .	24
6.2.1 The lemma . . . . .	24
6.2.2 Footnotes . . . . .	25
6.2.3 Endnotes . . . . .	25
6.2.4 Paragraph in critical apparatus . . . . .	27
6.2.5 Change lemma and line number . . . . .	27
6.2.6 Changing the names of commands for critical apparatus . . . . .	28
6.3 Disambiguation of identical words in the apparatus . . . . .	28
6.3.1 Basic use . . . . .	28

6.3.2 Notes about input encoding with UTF-8 processor . . . . .	29
6.3.3 Use with <code>\lemma</code> command . . . . .	29
6.3.4 Sameword for a group of words . . . . .	31
6.3.5 Customizing . . . . .	32
6.4 Apparatus of Manuscripts . . . . .	32
6.4.1 Marking sections of text . . . . .	33
6.4.2 Layout of the apparatus of manuscripts . . . . .	33
6.4.3 Settings . . . . .	34
6.5 Familiar notes . . . . .	34
6.5.1 Basic use . . . . .	34
6.5.2 Customizing mark . . . . .	34
6.5.3 Separator for multiple footnotes . . . . .	34
6.6 Printing the footnote mark without printing the footnote text . . . . .	35
6.7 Changing series . . . . .	35
6.7.1 Create a new series . . . . .	35
6.7.2 Delete series . . . . .	35
6.7.3 Series order . . . . .	35
6.8 Position of critical and familiar footnotes . . . . .	36
<b>7 Critical apparatus appearance</b> . . . . .	<b>36</b>
7.1 Notes arrangement in a series . . . . .	37
7.2 Control line number printing . . . . .	38
7.2.1 Print line number only at first time . . . . .	38
7.2.2 Print page number only at first time . . . . .	38
7.2.3 Arbitrary text before line number . . . . .	39
7.2.4 Separator for line range . . . . .	39
7.2.5 Abbreviate line range . . . . .	39
7.2.6 Disable line number . . . . .	40
7.2.7 Printing pstart number . . . . .	40
7.2.8 Printing stanza number . . . . .	40
7.2.9 Separator between line and subline numbers . . . . .	40
7.2.10 Separator between page and line numbers . . . . .	41
7.2.11 Space around number . . . . .	41
7.2.12 Space around line symbol . . . . .	41
7.2.13 Space in place of number . . . . .	41
7.2.14 Boxing line number and line symbol . . . . .	41
7.3 For endnotes . . . . .	42
7.4 Arbitrary code around line number . . . . .	42
7.5 Separator between the lemma and the note . . . . .	43
7.5.1 For footnotes . . . . .	43
7.5.2 For endnotes . . . . .	43
7.6 Font style . . . . .	44
7.6.1 For line number . . . . .	44
7.6.2 For the lemma . . . . .	44
7.6.3 For all notes . . . . .	44
7.7 Wrapping notes . . . . .	44

7.7.1 Wrapping lemmas . . . . .	44
7.7.2 Wrapping contents . . . . .	45
7.8 Indent of notes content . . . . .	45
7.9 Arbitrary code at the beginning of notes . . . . .	45
7.10 Arbitrary code before inserting note . . . . .	45
7.11 Options for footnotes in columns . . . . .	46
7.11.1 Alignment . . . . .	46
7.11.2 Size of the columns . . . . .	46
7.12 Options for paragraphed footnotes and notes grouped by line . . . . .	46
7.12.1 Mark separation of notes . . . . .	46
7.12.2 Ragged text . . . . .	47
7.13 Options for block of notes . . . . .	47
7.13.1 Grouping notes by line . . . . .	47
7.13.2 Text before notes . . . . .	47
7.13.3 Code before notes . . . . .	47
7.13.4 Spacing . . . . .	47
7.13.5 Rule . . . . .	48
7.13.6 Maximum height . . . . .	48
7.13.7 Width . . . . .	49
7.14 Footnotes and the <code>reledpar</code> columns . . . . .	49
7.15 Endnotes in one paragraph . . . . .	49
<b>8 Fonts</b>	<b>49</b>
<b>9 Verse</b>	<b>50</b>
9.1 Basic . . . . .	50
9.2 Define stanza indents . . . . .	50
9.3 Repeating stanza indents . . . . .	51
9.4 Manual stanza indent . . . . .	51
9.5 Stanza breaking . . . . .	51
9.6 Hanging symbol . . . . .	52
9.7 Long verse and page break . . . . .	52
9.8 Content before/after verses . . . . .	52
9.9 Numbering stanza . . . . .	53
9.10 Various tools . . . . .	54
9.11 Notes on empty lines . . . . .	54
<b>10 Grouping</b>	<b>54</b>
<b>11 Cross referencing</b>	<b>55</b>
11.1 Basic use . . . . .	55
11.2 Cross-referencing to a critical note . . . . .	55
11.3 Cross-referencing which return a number in any case . . . . .	55
11.3.1 Cross-referencing in order to define line number of a critical note . . . . .	56
11.4 Not automatic cross-referencing . . . . .	57
11.5 Normal $\LaTeX$ cross-referencing . . . . .	57

11.6 References to start and end lines . . . . .	57
11.6.1 Reference to main text lines . . . . .	57
11.6.2 References to lines that are commented on in the apparatus . . . . .	57
11.6.3 Settings . . . . .	58
11.7 Compatibility with <code>xr</code> package . . . . .	59
<b>12 Side notes</b>	<b>59</b>
12.1 Basics . . . . .	59
12.2 Setting . . . . .	60
12.2.1 Width . . . . .	60
12.2.2 Vertical position . . . . .	60
12.2.3 Distance to the main text . . . . .	60
12.2.4 Font . . . . .	60
12.2.5 Separator between notes . . . . .	61
<b>13 Indexing</b>	<b>61</b>
13.1 Basics . . . . .	61
13.2 Referring to critical notes . . . . .	61
13.3 Separator between page and line numbers . . . . .	62
13.4 Using <code>xindy</code> . . . . .	62
13.5 Advanced setting . . . . .	63
<b>14 Glossary</b>	<b>63</b>
14.1 Preamble setting . . . . .	63
14.2 Commands . . . . .	63
<b>15 Tabular material</b>	<b>63</b>
<b>16 Sectioning commands</b>	<b>67</b>
16.1 Sectioning commands without line numbers or critical notes . . . . .	67
16.2 Sectioning commands with line numbering and critical notes . . . . .	67
16.3 Optimization . . . . .	68
<b>17 Quotation environments</b>	<b>68</b>
<b>18 Page breaks</b>	<b>68</b>
18.1 Control page breaking . . . . .	68
18.2 Prevent page break in a long verses . . . . .	69
<b>19 Miscellaneous</b>	<b>69</b>
19.1 Known and suspected limitations . . . . .	70
19.1.1 Non-standard geometry . . . . .	70
19.1.2 <code>floatrow</code> package compatibility . . . . .	70
19.1.3 ‘No room for a new’ . . . . .	70
19.1.4 Marginal notes . . . . .	70
19.1.5 Paragraph shape . . . . .	70
19.1.6 Paragraphed footnotes . . . . .	71

19.1.7 Use with other packages . . . . .	71
19.2 Parallel typesetting . . . . .	72
<b>I Implementation overview</b>	<b>73</b>
<b>II Preliminaries</b>	<b>73</b>
II.1 Links with original edmac . . . . .	73
II.2 Package declaration . . . . .	73
II.3 Package options . . . . .	74
II.4 Loading packages . . . . .	76
II.5 Compatibility with LuaTeX . . . . .	76
II.6 Boolean flags . . . . .	76
II.7 Messages . . . . .	77
II.8 Gobbling . . . . .	84
II.9 Miscellaneous commands . . . . .	84
II.10 Prepare reledpar . . . . .	84
II.11 Booleans provided by other optional packages which are required in any case . . . . .	85
<b>III Sectioning commands</b>	<b>85</b>
<b>IV List macros</b>	<b>89</b>
<b>V Line counting</b>	<b>91</b>
V.1 Choosing the system of lineation . . . . .	91
V.2 Line number margin . . . . .	92
V.3 Line number initialization and increment . . . . .	93
V.4 Line number locking . . . . .	95
V.5 Line number style . . . . .	96
V.6 Line number printing . . . . .	96
V.7 Line number counters and lists . . . . .	97
V.8 Line number locking counter . . . . .	98
V.9 Line number associated to lemma . . . . .	98
V.10 Reading the line-list file . . . . .	102
V.11 Commands within the line-list file . . . . .	104
V.12 Writing to the line-list file . . . . .	117
<b>VI Marking text for notes</b>	<b>123</b>
VI.1 \edtext itself . . . . .	124
VI.2 Substitute lemma . . . . .	131
VI.3 Substitute line numbers . . . . .	132
VI.4 Lemma disambiguation . . . . .	133

<b>VII Paragraph decomposition and reassembly</b>	<b>139</b>
VII.1 Boxes, counters, \pstart and \pend . . . . .	139
VII.2 Processing one line . . . . .	145
VII.2.1 General process . . . . .	145
VII.2.2 Process for “normal” line . . . . .	146
VII.2.3 Process for line containing \eledsection command . . . . .	148
VII.2.4 Hooks . . . . .	149
VII.2.5 Sidenotes and marginal line number initialization . . . . .	149
<b>VIII Line and page number computation</b>	<b>150</b>
<b>IX Line number printing</b>	<b>153</b>
<b>X Pstart number printing in side</b>	<b>157</b>
<b>XI Restoring footnotes and penalties</b>	<b>158</b>
XI.1 Add insertions to the vertical list . . . . .	158
XI.2 Penalties . . . . .	160
XI.3 Printing leftover notes . . . . .	161
XI.4 Text before notes . . . . .	162
<b>XII Critical footnotes</b>	<b>163</b>
XII.1 Fonts . . . . .	163
XII.2 Individual note options . . . . .	163
XII.3 Notes language . . . . .	164
XII.4 General survey of the way we manage notes . . . . .	165
XII.5 General setup . . . . .	165
XII.6 Footnotes arrangement . . . . .	166
XII.6.1 User level macro . . . . .	166
XII.6.2 Normal footnote . . . . .	167
XII.6.3 Paragraphed footnotes . . . . .	172
XII.6.4 Columnar footnotes . . . . .	179
XII.7 Critical notes presentation . . . . .	186
XII.7.1 Font tools . . . . .	186
XII.7.2 Pstart number in footnote . . . . .	187
XII.7.3 Lemma printing . . . . .	187
XII.7.4 Line number printing . . . . .	188
XII.7.5 Footnote grouped by line . . . . .	196
<b>XIII Familiar footnotes</b>	<b>198</b>
XIII.1 Adjacent footnotes . . . . .	198
XIII.2 Regular footnotes for numbered texts . . . . .	199
XIII.3 Footnote formats . . . . .	201
XIII.4 Footnote arrangement . . . . .	202
XIII.4.1 User level macro . . . . .	202
XIII.4.2 Normal footnotes . . . . .	202
XIII.4.3 Two columns footnotes . . . . .	208

XIII.4.4 Three columns footnotes . . . . .	210
XIII.4.5 Paragraphed footnotes . . . . .	212
XIII.5 Wrapping footnote marks in hyperlink . . . . .	216
<b>XIV Code common to both critical and familiar footnote in normal arrangement</b>	<b>217</b>
<b>XV Footnotes' width for two columns</b>	<b>218</b>
<b>XVI Footnotes' order</b>	<b>219</b>
<b>XVII Footnotes' rule</b>	<b>219</b>
<b>XVIII Specific skip for first series of footnotes</b>	<b>220</b>
XVIII.0.1 Overview . . . . .	220
XVIII.0.2 User level command . . . . .	221
XVIII.0.3 Internal commands . . . . .	221
<b>XIX Endnotes</b>	<b>222</b>
XIX.1 Internal commands . . . . .	222
XIX.2 User level commands . . . . .	226
XIX.2.1 Inserting contents to endnotes . . . . .	226
XIX.2.2 Printing endnotes . . . . .	227
<b>XX Generate series of notes</b>	<b>233</b>
XX.1 Test if series is still existing . . . . .	234
XX.2 Init specific to <code>reledpar</code> . . . . .	234
XX.3 For critical footnotes . . . . .	234
XX.3.1 Options . . . . .	234
XX.3.2 Create inserts, needed to add notes in foot . . . . .	236
XX.3.3 Create commands for critical apparatus, <code>\Afootnote</code> , <code>\Bfootnote</code> etc. . . . .	236
XX.3.4 Set standard display . . . . .	239
XX.4 For familiar footnotes . . . . .	239
XX.4.1 Options . . . . .	239
XX.4.2 Create tools for familiar footnotes ( <code>\footnotex</code> ) . . . . .	240
XX.5 The endnotes . . . . .	242
XX.5.1 The auxiliary file . . . . .	242
XX.5.2 The main macro . . . . .	242
XX.5.3 Tools . . . . .	243
XX.5.4 Internal commands . . . . .	243
XX.5.5 The options . . . . .	244
XX.6 Init standards series (A,B,C,D,E) . . . . .	245



<b>XXI Setting series display</b>	<b>245</b>
XXI.1 Change series order . . . . .	245
XXI.2 Test series order . . . . .	246
XXI.2.1 Get the first series . . . . .	246
XXI.3 Series setting . . . . .	246
XXI.3.1 General way of working . . . . .	246
XXI.3.2 Tools to set options . . . . .	247
XXI.3.3 Tools to generate options commands . . . . .	248
XXI.3.4 Options for critical notes . . . . .	250
XXI.3.5 Options for familiar notes . . . . .	251
XXI.3.6 Options for endnotes . . . . .	252
XXI.4 Hooks for a particular footnote . . . . .	254
XXI.5 Alias . . . . .	254
<b>XXII Output routine</b>	<b>255</b>
XXII.1 Extra footnotes output . . . . .	255
XXII.2 Patching standard output's commands . . . . .	258
<b>XXIII Cross referencing</b>	<b>261</b>
XXIII.1 Compatibility with xref . . . . .	275
<b>XXIV Side notes</b>	<b>276</b>
<b>XXV Minipages and such</b>	<b>284</b>
<b>XXVI Indexing</b>	<b>289</b>
XXVI.1 Looking on package order . . . . .	289
XXVI.2 Auxiliary macros for \edindex . . . . .	289
XXVI.3 Code specific to \edindexin critical footnotes . . . . .	290
XXVI.4 Analysis of command in indexed text . . . . .	292
XXVI.5 Code for the formatted index . . . . .	292
XXVI.6 Main code . . . . .	293
XXVI.7 Hyperlink . . . . .	294
XXVI.8 'innote' and 'notenumber' option of indextols package . . . . .	297
<b>XXVII Glossaries</b>	<b>298</b>
<b>XXVIII Verse</b>	<b>300</b>
XXVIII.1 Hanging symbol management . . . . .	300
XXVIII.2 Using & character . . . . .	301
XXVIII.3 Code category setting . . . . .	301
XXVIII.4 Stanza count and indent . . . . .	301
XXVIII.5 Numbering stanza . . . . .	303
XXVIII.6 Stanza number in note . . . . .	304
XXVIII.7 Main work . . . . .	304
XXVIII.8 Restore catcode and penalties . . . . .	307

<b>XXIX Apparatus of Manuscripts</b>	<b>307</b>
XXIX.1 User level macro . . . . .	307
XXIX.2 Setting macro . . . . .	309
XXIX.3 Counters and lists . . . . .	309
XXIX.4 Auxiliary file macros . . . . .	310
XXIX.5 Action macro . . . . .	311
XXIX.6 Inserting footnote . . . . .	316
XXIX.7 Other . . . . .	316
<b>XXX Arrays and tables</b>	<b>316</b>
XXX.1 Preamble: macro as environment . . . . .	316
XXX.2 Tabular environments . . . . .	320
XXX.2.1 Disabling and restoring commands . . . . .	320
XXX.2.2 Counters, boxes and lengths . . . . .	323
XXX.2.3 Tabular typesetting . . . . .	327
XXX.2.4 Environments . . . . .	338
<b>XXXI Quotation's commands</b>	<b>338</b>
<b>XXXII Section's title commands</b>	<b>340</b>
XXXII.1 Commands to disable some feature . . . . .	340
XXXII.2 General overview . . . . .	340
XXXII.3 \beforeeledchapter command . . . . .	341
XXXII.4 Auxiliary commands . . . . .	341
XXXII.5 Patching standard commands . . . . .	342
XXXII.6 Main code of \eledxxx commands . . . . .	347
XXXII.7 Macros written in the auxiliary file . . . . .	350
<b>XXXIII Page breaking or no page breaking depending of specific lines</b>	<b>352</b>
<b>XXXIV Long verse: prevents being separated by a page break</b>	<b>354</b>
<b>XXXV Tools for hyperref package</b>	<b>354</b>
<b>XXXVI Compatibility with eledmac</b>	<b>355</b>
<b>Appendix A Things to do when changing versions</b>	<b>358</b>
Appendix A.1 Migrating from edmac to ledmac . . . . .	358
Appendix A.2 Migration from ledmac to eledmac . . . . .	359
Appendix A.3 Migration to eledmac 1.5.1 . . . . .	360
Appendix A.4 Migration to eledmac 1.12.0 . . . . .	360
Appendix A.5 Migration to eledmac 17.1 . . . . .	361
Appendix A.6 Migration to eledmac 1.21.0 . . . . .	361
Appendix A.6.1 \Xledsetnormalparstuff and \ledsetnormalparstuffX	361
Appendix A.6.2 Endnotes . . . . .	361
Appendix A.7 Migration to eledmac 1.22.0 . . . . .	361
Appendix A.8 Migration to eledmac 1.23.0 . . . . .	361

Appendix A.9 Migration from <code>eledmac</code> to <code>reledmac</code> . . . . .	362
Appendix A.9.1 Risk of ‘no room for a new’ . . . . .	362
Appendix A.9.2 Multiple indices with <code>memoir</code> . . . . .	362
Appendix A.9.3 Deprecated commands and options . . . . .	362
Appendix A.9.4 <code>\renewcommand</code> replaced by <code>command</code> . . . . .	363
Appendix A.9.5 Commands the names of which have been changed . . . . .	363
Appendix A.9.6 Endnotes . . . . .	365
Appendix A.9.7 Z Series . . . . .	365
Appendix A.9.8 Internal commands . . . . .	365
Appendix A.10 Migration to <code>reledmac</code> 2.1.0 . . . . .	365
Appendix A.11 Migration to <code>reledmac</code> 2.1.3 . . . . .	365
Appendix A.12 Migration to <code>reledmac</code> 2.3.0 . . . . .	365
Appendix A.13 Migration to <code>reledmac</code> 2.4.0 . . . . .	366
Appendix A.14 Migration to <code>reledmac</code> 2.5.0 . . . . .	366
Appendix A.15 Migration to <code>reledmac</code> 2.7.0 . . . . .	366
Appendix A.16 Migration to <code>reledmac</code> 2.7.2 . . . . .	366
Appendix A.17 Migration to <code>reledmac</code> 2.8.0 . . . . .	366
Appendix A.18 Migration to <code>reledmac</code> 2.13.1 . . . . .	366
Appendix A.19 Migration to <code>reledmac</code> 2.18.0 . . . . .	367
Appendix A.20 Migration to <code>reledmac</code> 2.21.0 . . . . .	367
<b>References</b>	<b>368</b>
<b>Index</b>	<b>368</b>
<b>Change History</b>	<b>416</b>

# 1 Introduction

## 1.1 Aim of the package

The `reledmac` package, together with  $\text{\LaTeX}$ , 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 to both prose and verse;
- multiple series of footnotes and endnotes;
- block or columnar formatting of the footnotes;
- simple tabular material may be line numbered;
- indexing keyed to page and line numbers.

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

Apart from `reledmac` there are other  $\LaTeX$  packages for typesetting critical editions. However, the aim of `reledmac` is to provide an “all in one” and flexible tool in the field of critical editions.

Any suggestions for new features are welcome.

This manual contains a general description of how to use `reledmac` followed by the complete source code and its extensive documentation (in sections I and following, enumerated with Roman numerals). It ends with a list of actions to do when migrating from one version to other, a change history and an index to the source code.

You do not 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 earlier sections. But no documentation, however thorough, can cover every question that comes up and many can be answered quickly by consulting the code. On a first reading, we suggest that you 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  $\TeX$  format for critical editions’, *TUGboat* 11 (1990), pp. 623–643.

From 1991 through 1994, the macros continued to evolve, and were tested at a number of sites. We are very grateful to all the members of the (now defunct) `edmac@mailbase.ac.uk` discussion group who helped us with smoothing out the bugs and infelicities in the macros. Ron Whitney and our anonymous reviewer at the TUG

<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).

were both of great help in ironing out last-minute wrinkles, while Ron made some important suggestions which may help to make future versions of `edmac` even more efficient. Wayne Sullivan, in particular, provided several important fixes and contributions, including adapting the Mittelbach/Schöpf ‘New Font Selection Scheme’ for use with `PLAIN TEX` and `edmac`. Another project Wayne has worked on is a DVI post-processor which works with an `edmac` that has been slightly modified to output `\specials`. This combination enables you to recover to some extent the text of each line as ASCII code, facilitating the creation of concordances, an *index verborum*, etc.

As of 1994, we were pleased to be able to say that `edmac` was being used for the 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 *Rithmarchia* 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.

### 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

<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 *Rithmarchia* des Werinher von Tegernsee’, *ibid.*

<sup>8</sup>Geert H. M. Claassens, *De Middelnederlandse Kruisvaartromans*, (Amsterdam: Schipphower 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. Gádeke 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.

February 2004. Sidenotes and regular footnotes in numbered text were added in April 2004. This port was called `ledmac` ( $\text{\LaTeX}$  `edmac`).

Since July 2011, `ledmac` is maintained by Maïeul Rouquette. It is increasingly powerful and flexible, but it also has become increasingly divergent from the original  $\text{\TeX}$  macro.

### 1.2.3 `eledmac`

Important changes were put in version 1.0, to make `ledmac` more easily extensible (see 7 p. 36). 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 A.2 p. 359.

### 1.2.4 `reledmac`

`eledmac` has facilitated the creation of customized critical editions. However, the changes made to allow such customization were made in a non-systematic way. Many deprecated commands were kept and many technical ‘debts’ were accumulated, hindering the future evolution of the package.

For these reasons, Maïeul Rouquette decided on a spring cleaning of the code. As some commands name were changed, the resulting compatibility was broken (a little).

A new name was selected: `reledmac` (extended renewed `eledmac`). To migrate from `eledmac` to `reledmac`, please read Appendix A.9 p. 362.

## 1.3 Bibliography

A collaborative list of works edited with (r)(e)ledmac is available at [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.

If you write book or article about (r)(e)ledmac, please add it on the collaborative bibliography on [https://www.zotero.org/groups/articles\\_and\\_books\\_about\\_reledmac/items](https://www.zotero.org/groups/articles_and_books_about_reledmac/items).

## 2 How the package works — the problem of the number of $\text{\LaTeX}$ runs

The `reledmac` package is a three-pass package like  $\text{\LaTeX}$  itself. Although your textual apparatus and line numbers will be printed on the first run, it takes two more compilations by  $\text{\LaTeX}$  to be sure that everything is correctly placed, and one more if you typeset right-to-left text with  $\text{\XeLaTeX}$ . If you make any subsequent changes altering the number of lines or notes, the input file may similarly require three passes to get everything to the right place. `reledmac` will tell you that you need to make more runs when it detects changes, 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  $\text{\LaTeX}$  once or twice more.

However, the best way to be sure that one has made the right number of runs is to use some of  $\text{\LaTeX}$ 's run scripts like *latexmk*.

### 3 Compatibility warning

If you use other classes than `\article` or `\book`, or modify the layout with `geometry`, some settings should be made to have correct height for the blocks of notes.

Please read 7.13.6 p. 48.

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.

## 4 Options

The package can be loaded with a number of global options which are listed here. There are two types of options: 1) options which provide specific features, and, 2) options which optimize the package's performance. It is advisable for you to read the relevant parts of the handbook, before reading about the first type of option (specific features), but you can look at the second type (package optimization) in your first reading of the manual.

### 4.1 Specific features

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

**auxdir** `reledmac` generates auxiliary files. It could be useful to store them in a specific directory. You can set it using `auxdir=<folder>` option. Note the two following point:

1.  $\text{\TeX}$  is not able to create folder. You should create it yourself.
2. The option does not change the default  $\text{\LaTeX}$  auxiliary files (`.aux`, `.toc`, ...).

**eledmac-compat** help to migrate from `eledmac` to `reledmac` (see Appendix A.9.5 p. 363).

**nopenalties** must be called in some cases when using paragraphed endnotes (?? p. ??)

**nopbinverse** prevents page break within verse environment.

**noquotation** by default, the quotation environment is redefined within numbered text. You can disable this redefinition with `noquotation` (see 17 p. 68).

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

**widthliketwocolumns** set the width of the text printed in a single column to be the same as the width of the text printed in two parallel columns with `reledpar`. This is useful when alternating between normal and parallel typesetting.

**xindy** and `xindy+hyperref` select `xindy` as the index processor (13.4 p. 62).

## 4.2 Optimizing package 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 (16.2 p. 67).

**noend** disables tools for endnotes (`\Aendnote`, `\Bendnote` etc.). If you do not need endnotes, this option lets `reledmac` 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 use of a series of critical notes and a new series of normal notes inside `minipage` and `ledgroup` environments (see 10 p. 54). 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 the entire 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.

## 5 Text lines and paragraphs numbering

### 5.1 Text lines numbering

`\beginnumbering` Each section of numbered text must be preceded by `\beginnumbering` and followed by `\endnumbering`, as in the following example.

```
\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).

## 5.2 Paragraphs

### 5.2.1 Basics

`\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

```

### 5.2.2 Automatically producing \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>

### 5.2.3 Content before specific \pstart and after specific \pend

Both `\pstart` and `\pend` can take a 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`.

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

Note that a `\noindent` will be automatically added before this argument, and, consequently, a `\parskip` will be inserted. You can use a second optional argument, in brackets, to not have this `\noindent`.

```
\pstart[foo] % A \noindent will be inserted before foo.
\pstart[] [foo]% No \noindent before foo.
```

The second optional argument of `\pstart` / `\pend` replace the argument of `\AtEveryPstart*` / `\AtEveryPend*`.

If you need to start a `\pstart` with 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 9 p. 50) or `reledpar` (see 19.2 p. 72).

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

#### 5.2.4 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`

Note that a `\noindent` will be inserted before the argument, and, consequently, a `\parskip`. You can use the starred version of `\AtEveryPstart` and `\AtEveryPend` to no insert the `\noindent`.

`\AtStartEveryPstart`    The argument of `\AtStartEveryPstart` / `\AtEndEveryPend` will be inserted at  
`\AtEndEveryPend` the beginning / the end of every `\pstart` / `\pend` in the same paragraph. For example, if you want each `\pstart` to start with a star, you can use:

```
\AtStartEveryPstart{*}
```

Instead of manually doing

```
\pstart * Real pstart content.\pend
```

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

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

```
\setcounter{pstart}{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`.

### 5.2.6 Languages written in Right to Left

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

### 5.2.7 Memory limits

**This paragraph is kept for history, but the problems described below should not appear with the most recent version of  $\TeX$ .**

`\pausenumbering`  
`\resumenumbering` 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 solution 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 type,  
`\newcommand{\memorybreak}{\pausenumbering\resumenumbering}`

and type `\memorybreak` between the relevant `\pend` and `\pstart`.

## 5.3 Lineation commands

### 5.3.1 Disabling lineation

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

### 5.3.2 Setting lineation start and step

`\firstlinenum` By default, reledmac numbers every 5th line. There are two counters that control this behaviour: `firstlinenum` and `linenumincrement`. 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:

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

`\linenumberlist` to have numbers printed on prime-numbered lines only. There must be no spaces within the definition which consists of comma-separated integer 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 empty definition

```
\gdef\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.

### 5.3.3 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.

### 5.3.4 Setting line number margin

`\linenummargin` The command `\linenummargin{<location>}` specifies the margin where the line (or `pstart`) numbers will be printed. The permissible values for `<location>` are `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 the value in effect at the `\pend` of a numbered paragraph. Apart from an initial setting for `\linenummargin`, only change `\linenummargin` 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 of the current paragraph).

### 5.3.5 Other settings

`\leftlinenum` When a marginal line number is to be printed, there are many ways to display it. You can  
`\rightlinenum` redefine `\leftlinenum` and `\rightlinenum` to change the way marginal line numbers  
`\linenumsep` 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.

## 5.4 Changing the line numbers

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

### 5.4.1 Sublineation

`\startsub` You insert the `\startsub` and `\endsub` commands in your text to turn sub-lineation  
`\endsub` on and off. For example, stage directions in plays 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 it changes in the middle.

You can change the separator between line number and subline number either by using `\Xsublinesep` without any optional argument (7.2.9 p. 40) or by using `\Xsublinesepside`. But in the second case, it will change the separator only for line numbers in the margins, not in the footnotes.

### 5.4.2 Locking lineation

`\startlock` The `\startlock` command, used in running text, locks the line number at its current  
`\endlock` value, until you insert `\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. But in this case you may use the `\stanza` mechanism, see 9 p. 50.

`\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, assuming that the settings of the previous parameters requires the display of 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}`.

### 5.4.3 Setting and changing line number

`\setline` In some cases you may want to modify the line numbers that are automatically calculated.  
`\advanceline`

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

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

#### 5.4.4 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.

#### 5.4.5 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.

`\hidenumberingonleftpage` `\hidenumberingonleftpage` is like `hidenumbering`, but is applied only on left page. `\hidenumberingonrightpage` is applied on right page. They can be useful if the position of the line number is depending of the position of the page, but the position of marginal note is fixed.

## 5.5 Executing code at each line

`\dolinehook`  
`\doinsidelinehook`

`reledmac` provides an advanced feature for users. The argument passed to `\dolinehook{⟨arg⟩}` 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, when the line number has already been fixed. In many cases, the latter is more useful than the former. The file `examples/2-line_numbers_in_header.tex` provides an example for printing the first and last line numbers of a page in the header.

## 6 Apparatus commands

### 6.1 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 familiar footnote.

So :

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

### 6.2 Critical notes

#### 6.2.1 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.		
		<hr style="width: 20%; margin-left: 0;"/>
	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:

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

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

### 6.2.2 Footnotes

The second argument of the `\edtext` macro, `\commands`, 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{\text}`. When all of the six are used, the A notes appear in a layer just below the main text, followed by the rest in turn, down to the E notes at the bottom. These are the main macros that you will use to construct the critical apparatus of your text.

If you need more series of critical notes, please look at 6.7.1 p. 35.

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. 7.2.5 p. 39).
- `nonum` disables line numbering for this note. A horizontal blank space is added instead. You can use `\Xinplaceoflemmaseparator` to set it (7.5.1 p. 43).
- `nosep` to disable the lemma separator for this note.
- `linerangesep=<c>` to change to `<c>` the separator between start line and end line for this particular note.

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

### 6.2.3 Endnotes

`\Aendnote` **Inserting endnotes** 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 `reledmac`.

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

`\Aendnote[option]{\text}`.

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

- `fulllines` to disable `\Xendtwolines` and `\Xendmorethantwolines` features for this particular note (cf. 7.2.5 p. 39).
- `nonum` to disable line number for this particular note.
- `nosep` to disable the lemma separator for this particular note. A horizontal blank space is added instead. You can use `\Xendinplaceoflemmaseparator` to set it (7.5.2 p. 43).
- `linangesep= $\langle c \rangle$`  to change to  $\langle c \rangle$  the separator between start line and end line for this particular note.

**\doendnotes** **Printing endnotes** 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 sections.

**\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{ $\langle s \rangle$ }`. For each value of  $\langle s \rangle$ , 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 (7.5.2 p. 43).

As endnotes may be printed at any point in the document they always start with the page number where they are called.

**toendnotes** **Code between endnotes** Sometimes, it is useful to insert content between endnotes of the same series: for example to separate endnotes of different sections of the same text. In this case, you could use *inside numbered text* the command:

`\toendnotes[ $\langle series \rangle$ ]{ $\langle content \rangle$ }` where  $\langle series \rangle$  is a comma-separated list of the series of endnotes where  $\langle content \rangle$  must be inserted. If  $\langle series \rangle$  is empty, then  $\langle content \rangle$  is inserted to all the series.

For example:

```
\toendnotes{\section{Section's title}}
```

Alternatively, you can use `\Xtoendnotes{⟨content⟩}`, where “X” must be replaced by a series letter.

Remember that the endnotes are temporarily stored in an auxiliary file. That means in general you want to write the `⟨content⟩` in the auxiliary file *without expanding it*, that is without interpreting TeX content.

However, in some cases, you may want to write a once-expanded<sup>14</sup> version of the `⟨content⟩`, that is the version where the commands are expanded on the first level. This can be, for example, to get a counter value. Use the starred version in this case. For example:

```
\Attoendnotes*{\string\section{Letter 1 (chap. \thechapter)}}
```

#### 6.2.4 Paragraph in critical apparatus

By default, no paragraph can be made in the notes of the critical apparatus. You can allow it by adding the options `parapparatus` when loading the package :

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

Note that you *cannot* use paragraphs (e.g. blank lines or `\par`) inside of notes, when they are set to paragraph arrangement!

#### 6.2.5 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` and 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          1 I am happy : I saw my friend Smith on
  \edtext{Smith}{\Afootnote{Jones 2 Tuesday.
    C, D.}} on Tuesday.}
{\lemma{I \dots\ Tuesday.}
  \Bfootnote{The date was
    July 16, 1954.}
}
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. `⟨arg⟩` actually consist of 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). I.e.

<sup>14</sup>The expansion mechanism of TeX is a quite complex problem, but fundamental. We have no place to explain it fully here. Read introduction to TeX to understand well.

`\linenum{⟨start page⟩|⟨s. line⟩|⟨s. sub-l.⟩|⟨end p.⟩|⟨e. l.⟩|⟨e. sub-l.⟩|⟨font⟩|}`

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 only 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 notes. 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 (11 p. 55) 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.

### 6.2.6 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 type 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>15</sup>

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

## 6.3 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.

### 6.3.1 Basic use

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

<sup>15</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

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.

### 6.3.2 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 (code positions) 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>16</sup> with the `buffer` option set to `true`.

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

### 6.3.3 Use with `\lemma` command

If you use the `\lemma` command, `reledmac` 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}}
```

---

handbook of `xargs` to know more about `\newcommandx`.

<sup>16</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}}.%

```

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

Consequently, you must tell reledmac to which instance of `\sameword` you are referring in the first argument of `\edtext`:

- In the content of `\lemma`, use `\sameword` with no optional argument.
- In the first argument of `\edtext`, use `\sameword` with the optional argument  $\langle X \rangle$ .  $\langle X \rangle$  is the depth of the `\edtext` where the `\lemma` is used. So if the `\lemma` is called in a `\edtext` inside another `\edtext`,  $\langle X \rangle$  is equal to 2. If the `\lemma` is called in a `\edtext` “of first level”,  $\langle X \rangle$  is equal to 1. If the lemma is called in both 1 and 2 `\edtext` depth,  $\langle X \rangle$  is 1,2. If that word is referenced in the lemma of every `\edtext` depth,  $\langle X \rangle$  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  $\langle X \rangle$  does not refer to the level where the `\sameword` occurs, but to the level of the `\lemma` that refers to that `\sameword`. For example:

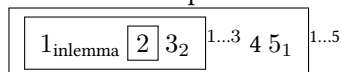
```

\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 example figure, 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 1 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.

Here, the corresponding code:

```
\edtext{%
  \edtext{%
    \sameword[inlemma]{A} (1)
    \edtext{%
      \sameword{A} (2)
    }%
  }%
  \Afootnote{level~3}%
}%
\sameword[2]{A} (3)
}%
{%
  \lemma{%
    \sameword{A}%
    \ldots%
    \sameword{A}%
  }%
  \Afootnote{level~2}%
}%
\sameword{A} (4)
\sameword[1]{A} (5)
}%
{%
  \lemma{\sameword{A}\ldots\sameword{A}}%
  \Afootnote{level~1}%
}%
}
```

1	A (1) A (2) A (3) A (4) A (5)
<hr style="width: 50%; margin-left: 0;"/>	
1	A <sup>1</sup> ...A <sup>5</sup> ] level 1
1	A <sup>1</sup> ...A <sup>3</sup> ] level 2
1	A <sup>2</sup> (2) ] level 3

#### 6.3.4 Sameword for a group of words

Sometime, a group of words, and not only a single word, occurs multiple times. In this case, you have two solutions.

First, you can consider only individual words, and not groups of word. See this example:

```
\sameword{per} \sameword{causam}
tamen scire
\edtext{\sameword{causam}}{\Bfootnote{fnote}}
est
\edtext{\sameword{per} \sameword{causam}}{\Bfootnote{causam rei B}}
cognoscere
\edtext{\sameword{causam}}{\Bfootnote{fnote}}
```

1 per causam tamen scire causam est per causam cognoscere causam

---

1 causam<sup>2</sup> ] fnote

1 per<sup>2</sup> causam<sup>3</sup>] causam rei B

1 causam<sup>4</sup>] fnote

Here, there no ambiguity about which “per causam” we speak.

However, we may consider that as “per causam” is the lemma of the second note, there should be only one number for the whole lemma. In this case we can mark all the “per causam” groups. But as “causam” is also called as lemma in note 1 and 3, we need to use nested `\sameword`. Consequently, we need to use `\lemma` for the `\edtext` linked to “per causam”, as we don’t want to have number for individual word.

```
\sameword{per \sameword{causam}} tamen scire
\edtext{\sameword{causam}}{\Bfootnote{fnote}} est
\edtext{\sameword[1]{per \sameword{causam}}}{\lemma{\sameword{per causam}}\Bfootnote{causam}}
\edtext{\sameword{causam}}{\Bfootnote{fnote}}
```

1 per causam tamen scire causam est per causam cognoscere causam

---

1 causam<sup>2</sup>] fnote

1 per causam<sup>2</sup>] causam rei B

1 causam<sup>4</sup>] fnote

### 6.3.5 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}%
}
```

## 6.4 Apparatus of Manuscripts

The critical notes mostly refer to textual variants between manuscripts which contain the text to be edited. It may so happen that the manuscripts only contain parts of the text. Depending on one’s wishes, `reledmac` can generate lists of relevant manuscripts for any delimited portion of text. Such lists are referred to as “apparatuses of manuscripts”.

To produce an apparatus of manuscripts with `reledmac`, you have to insert specific commands that are used to mark the sections for which only part of the manuscripts are relevant. These commands will be processed, and **after the second T<sub>E</sub>X run**, corresponding apparatuses of manuscripts will be inserted in the first (viz. ‘A’ series) level of footnotes.

As the insertion of this apparatus can change the page breaks, you may have to run T<sub>E</sub>X two or more times. We strongly recommend to use tools like *latexmk* to do that.



### 6.4.1 Marking sections of text

`\msdata` `\msdata{⟨text⟩}` must be inserted at the point where a section for which only part of the manuscripts are relevant starts. `⟨text⟩` can be any arbitrary text, viz. a list of the manuscripts that are used for the section that starts. The command must be attached right at the point where the section starts, with no space, like so:

```
\msdata{ABC}Lorem ipsum
```

Which means that the section of text starting by “Lorem ipsum” is witnessed by manuscripts A, B and C.

`\stopmsdata` `\stopmsdata` must be inserted at the point where the section of text previously marked by `\msdata` ends. The command must be attached right to the end of the section, with no space. As `\stopmsdata` is a  $\TeX$  argumentless macro, it will gobble the following space. To keep that space, you have to either append a backslash followed by a space or `{}` to `\stopmsdata`, like so:

```
\msdata{ABC}Lorem ipsum dolor
[...]
amet\stopmsdata{} \msdata{ABCD}sic transit [...]
```

Which means that the part of text containing “Lorem ipsum dolor ... amet” is witnessed by manuscripts A, B and C, while the part of text starting by “sic transit” is witnessed by manuscripts A, B, C and D.

`\stopmsdata` is also automatically inserted by `\msdata`.

Note that in most cases, any `\stopmsdata` is followed by `\msdata`. However, as these two command are usually separated by a space, it may happen that a line break be automatically inserted between them. This is why it is advised to always insert `\stopmsdata`, even if `\msdata` inserts it in case it is forgotten.

### 6.4.2 Layout of the apparatus of manuscripts

On every page, the apparatus of manuscripts marks the corresponding section with starting and ending line numbers. However, the following rules will be applied:

- If the section does not start on the current page, the starting line number will be the line number of the first line on the page.
- If the section does not stop on the current page, the ending line number will be the line number of the last line on the page.
- If the section neither starts nor ends on the current page, no line number will be printed. The same is true in case both `\msdata` is called at the very beginning of the page and `\endmsdata` is called at the very end of the page.

### 6.4.3 Settings

As the apparatus of manuscripts technically consists of first-level critical notes ('A' series), any setting available for critical notes can be applied (7 p. 36). However, the following *additional* commands are available.

<code>\setmsdataseries</code>	The series used by default for the apparatus of manuscripts is series A. However, you can change it with <code>\setmsdataseries{&lt;series&gt;}</code> .
<code>\setmsdatalabel</code>	As the apparatus of manuscripts consists of regular critical footnotes, a lemma is associated to them. By default, it is "Ms.". You can change it using <code>\setmsdatalabel{&lt;txt&gt;}</code> .
<code>\setmsdataposition</code>	If you want the manuscript apparatus to be on the same level of critical footnotes as the other apparatuses, for each line, reledmac will first insert the manuscript apparatus, then the other footnotes. You can change it using: <code>\msdataposition{regular-msdata}</code> And restore the default behaviour using <code>\msdataposition{msdata-regular}</code>

## 6.5 Familiar notes

### 6.5.1 Basic use

<code>\footnoteA</code>	As well as the standard L <sup>A</sup> T <sub>E</sub> X footnotes generated via <code>\footnote</code> , the package also provides five series of additional footnotes called <code>\footnoteA</code> through <code>\footnoteE</code> . These have the familiar marker in the text, and the marked text at the foot of the page can be formatted using any of the styles described for the critical footnotes. Note that the 'regular' footnotes have the series letter at the end of the macro name whereas the critical footnotes have the series letter at the start of the name.
<code>\footnoteB</code>	
<code>\footnoteC</code>	
<code>\footnoteD</code>	
<code>\footnoteE</code>	

### 6.5.2 Customizing mark

<code>\thefootnoteA</code>	Each series uses a set of macros for styling the marks. The mark numbering scheme of series A is defined by the <code>\thefootnoteA</code> macro; the default is: <code>\renewcommand*{\thefootnoteA}{\arabic{footnoteA}}</code> The appearance of the mark in the text is controlled by <code>\bodyfootmarkA</code> which is defined as: <code>\newcommand*{\bodyfootmarkA}{%</code> <code>\hbox{\textsuperscript{\normalfont\@nameuse{@thefnmarkA}}}</code> The command <code>\footfootmarkA</code> controls the appearance of the mark at the start of the footnote text. It is defined as: <code>\newcommand*{\footfootmarkA}{\textsuperscript{\@nameuse{@thefnmarkA}}}</code> There are similar command triples for the other series.
<code>\bodyfootmarkA</code>	
<code>\footfootmarkA</code>	

### 6.5.3 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 reledmac provides this automatically.

<code>\multfootsep</code>	<code>\multfootsep</code> is used as the separator between footnote markers. Its default definition is:
---------------------------	---

`\providecommand*{\multfootsep}{\textsuperscript{\normalfont,}}`  
and can be changed if necessary.

## 6.6 Printing the footnote mark without printing the footnote text

`\footnoteXmark`  
`\footnoteXtext`

In certain cases, you can't directly use `\footnoteX`; for example, when using `\uline` command of the `ulem` package. You need to print the footnote mark first, then call the footnote text to be inserted.

For all  $\langle X \rangle$  command, `reledmac` provides a `\footnote\langle X \rangle mark` command and a `\footnote\langle X \rangle text` command, equivalent to standard  $\text{\LaTeX}$ 's command `\footnotemark` and `\footnotetext`. For example, to use with `\uline`, do:

```
This is \uline{a test containing\mbox{\footnoteAmark}}\footnoteAtext{A
simple footnote.}\uline{ a simple footnote.}
```

If you use `reledpar`, you can't use these two commands to print the footnote mark on one side and the footnote text on the other side.

You must use `\footnote\langle X \rangle nomk` and `\footnote\langle X \rangle mk`, defined in `reledpar` (?? p. ??)

## 6.7 Changing series

### 6.7.1 Create a new series

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

### 6.7.2 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–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}
```

### 6.7.3 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.

`\seriesatbegin`  
`\seriesatend`

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

## 6.8 Position of critical and familiar footnotes

`\fnpos`    There is a historical incoherence in (r)(e)ledmac. The familiar footnotes are before  
`\mpfnpos` 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}
```

You can also decide to alternate familiar and critical footnotes with your own order.  
In this case, the second argument of `\fnpos` or `\mpfnpos` is a comma separated list of  
values. Each value has the following form:

$\langle series \rangle \langle type \rangle$

$\langle series \rangle$  is a series letter (A,B,C etc.), while  $\langle type \rangle$  must be either “critical” or “familiar”.

For example, suppose you want to first print the familiar footnotes of the “A” series,  
then all the series of critical footnotes, and finally all the series of familiar footnotes,  
except the “A” series. In this case, use the following command:

```
\fnpos{%
  {A}{familiar},
  {A}{critical},%
  {B}{critical},%
  {C}{critical},%
  {D}{critical},%
  {E}{critical},%
  {B}{familiar},%
  {C}{familiar},%
  {D}{familiar},%
  {E}{familiar}%
}
```

Note that you must define the position of all the series of footnotes you use. If you don’t,  
you will have infinite runs of  $\LaTeX$ .

## 7 Critical apparatus appearance

Some commands can be used to change the display of the footnotes. All can have an  
optional argument [ $\langle s \rangle$ ], 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 apply to the entire series.

When a length, noted  $\langle l \rangle$ , is used, it can be stretchable:  $a$  plus  $b$  minus  $c$ . The final length  $m$  is calculated by  $\text{\TeX}$  to have:  $a - c \leq m \leq a + b$ . If you use some relative unit<sup>17</sup>, it will be relative to font size of the footnote, except for commands concerning the place kept by the notes — including blank space.

Some commands are boolean, indicating when an option is enabled. If you want to disable the option after enabling it, you must use `[false]` as the second optional argument. For example:

- `\XX[A][false]` to disable the ‘XX’ option for the series A.
- `\XX[] [false]` to disable it for all series.

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.

## 7.1 Notes arrangement in a series

`\Xarrangement`  
`\arrangementX`

By default, all footnotes are formatted as a series of separate paragraphs in one column. 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 of the footnotes of a series as a single paragraph; if you use this arrangement, you are strongly encouraged to read 19.1.6 p. 71.
- `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.

Note that you *cannot* use paragraphs (e.g. blank lines or `\par`) or line breaks (`\break` or `\linebreak` or `\newline` etc.) inside of notes, when they are set to paragraph arrangement!

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 note arrangement again.

<sup>17</sup>Like `em` which is the width of an ‘m’ in a given font.

`\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 the arrangement macro again afterwards to take account of the new value.

## 7.2 Control line number printing

### 7.2.1 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[⟨s⟩]`.

`\Xnumberonlyfirstintwolines` 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 `\Xnumberonlyfirstintwolines[⟨s⟩]`, a distinction is made.

`\Xsymlinenum` 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 non-robust command.

`\Xendnumberonlyfirstinline` For endnotes, `\Xendnumberonlyfirstinline`; `\Xendnumberonlyfirstintwolines`  
`\Xendnumberonlyfirstintwolines` and `\Xendsymlinenum` are the equivalents of  
`\Xendsymlinenum` `\Xnumberonlyfirstinline`; `\Xnumberonlyfirstintwolines` and `\Xsymlinenum`.

### 7.2.2 Print page number only at first time

For endnotes, `reledmac` provides a mechanism for printing the page number only the first time it is seen. However, when a lemma spans over two pages, the line numbers are normally printed in the following pattern: starting page number - starting line number - ending page number - ending line number. It follows that what corresponds to the actual ‘page number’ may not be self-evident. So: `\Xendpagenumberonlyfirst[⟨s⟩]` can be called to ensure that the starting page number of a lemma be not printed if it is the same as the ending page number of the preceding lemma. You can use *additionally* one (and only one) of the following commands:

- `\Xendpagenumberonlyfirstifsingle[⟨s⟩]`: the first page number of the lemma will not be printed only if the following conditions are true:
  1. The starting page number of the lemma is the same as the ending page number of the preceding lemma.
  2. The ending page number of the lemma is the same as the starting page number of the lemma.

In this case the ending page number will always be printed if it is different from the starting page number.

- `\Xendpagenumberonlyfirstintwo[⟨s⟩]`: both the starting page number and the ending page number of a lemma are not printed if they are both the same as

`\Xendpagenumberonlyfirst`

the starting page number and the ending page number of the preceding lemma respectively.

In any case, you can use:

- `\Xendsympagenum` • `\Xendsympagenum[⟨series⟩]{⟨c⟩}` to print `⟨c⟩` when the page number is not printed.
- `\Xendinplaceofpagenumber` • `\Xendinplaceofpagenumber[⟨series⟩]{⟨l⟩}` to print a `⟨l⟩` length horizontal space in case no symbol is printed instead of the page number.

### 7.2.3 Arbitrary text before line number

- `\Xbeforenumber` `\Xbeforenumber[⟨s⟩]{⟨txt⟩}` allow to insert `⟨txt⟩` before the line number, only when the line number is printed, so taking into account `\Xnumberonlyfirstinline` and similar.

### 7.2.4 Separator for line range

- `\Xlinerangeseparator` By default, the separator between the begin line and the end line in a lines' range is an en-dash in a normal font (`\textnormal{--}`). You can change it for critical footnotes with `\Xlinerangeseparator[⟨s⟩]{⟨text⟩}`, and with `\Xendlinerangeseparator[⟨s⟩]{⟨text⟩}` for critical endnotes.

### 7.2.5 Abbreviate line range

- `\Xtwolines` If a lemma is printed on two subsequent lines, `reledmac` 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)”.
- `\Xmorethantwolines`

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

```
\Xtwolines{sq.}
\Xmorethantwolines{sqq.}
```

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

If you use `\Xtwolines` without setting `\Xmorethantwolines`, the `⟨text⟩` argument of `\Xtwolines` 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 `\Xtwolinesbutnotmore[⟨series⟩]`.

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 `\Xendtwolines` symbol.

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

You can disable `\Xtwolines` and related for a specific note by using the ‘`[fulllines]`’ argument in the note macro cf. 6.2.2 p. 25.

For endnotes, use these macros: `\Xendtwolines`; `\Xendmoreethantwolines`; `\Xendtwolinesbutnotmore`; `\Xendtwolinesonlyinsamepage` instead of `\Xtwolines`; `\Xmoreethantwolines`; `\Xtwolinesbutnotmore`; `\Xtwolinesonlyinsamepage`.

### 7.2.6 Disable line number

`\Xnonumber` You can use `\Xnonumber[⟨s⟩]` if you do not want to have the line number in a footnote.  
`\Xendnonumber` `\Xendnonumber[⟨s⟩]` is the same for endnote.

### 7.2.7 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. 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.

### 7.2.8 Printing stanza number

`\Xstanza` You can use `\Xstanza[⟨s⟩]` if you want to print the stanza number in the footnote, before the line and subline number.

Of course the stanza number is printed only when you use `\numberstanza`

`\Xstanzaseparator` When using `\Xstanza`, you can use `\Xstanzaseparator[⟨s⟩]{⟨text⟩}` to print `⟨text⟩` after the stanza number. Default value is empty.

### 7.2.9 Separator between line and subline numbers

`\Xsublinesep` `\Xsublinesep[⟨s⟩]{⟨txt⟩}` changes the separator between line and subline in footnotes.

**Employed without optional argument, it also change separator in side number.**

`\Xendsublinesep` `\Xendsublinesep[⟨s⟩]{⟨txt⟩}` does the same thing for endnotes.

**However, it does not change anything for the separator in side number. Use `\Xsublinesep` without optional argument or `\Xsublinesepside{⟨txt⟩}` to do it.**  
 The default value is `\textnormal{.}`.



### 7.2.10 Separator between page and line numbers

`\Xpagelinesep` `\Xpagelinesep[⟨s⟩]{⟨txt⟩}` changes the separator between the page and line number in footnotes.

By default, the value defined for `\Xsublinesep` is used.

### 7.2.11 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.

`\Xendbeforenumber` `\Xendbeforenumber` and `\Xendafternumber` are the equivalents of `\Xbeforenumber` and `\Xafternumber` for endnotes.

`\Xnonbreakableafternumber` By default, the space defined by `\Xafternumber` is breakable. With `\Xnonbreakableafternumber[⟨s⟩]` it becomes nonbreakable.

### 7.2.12 Space around line symbol

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

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

`\Xendbeforemsymlinum` `\Xendbeforemsymlinum` and `\Xendaftersymlinum` are the equivalents of `\Xbeforemsymlinum` and `\Xaftersymlinum` for the endnotes.

### 7.2.13 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.

### 7.2.14 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}
```

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

`\Xendboxsymlinenum` `\Xendboxsymlinenum[⟨s⟩]{⟨l⟩}` is the same as `\Xboxsymlinenum` but for endnotes.

`\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.

### 7.3 For endnotes

`\Xendbeforepagenumber` `\Xendbeforepagenumber[⟨s⟩]{⟨text⟩}` defines the text before the page number in endnotes. Default value is `p.` (“p” followed by a dot).

`\Xendafterpagenumber` `\Xendafterpagenumber[⟨s⟩]{⟨text⟩}` defines the text after the page number in endnotes. Default value is `)` (open parenthesis followed by a single space). `\Xendlineprefixsingle[⟨s⟩]` defines the text before the line number in endnotes, when there is only one line. Default value is empty. `\Xendlineprefixmore[⟨s⟩]{⟨text⟩}` defines the text before the line number in endnotes, when there is more than one line. Default value is empty. If you don’t define it, use the value defined by `\Xendlineprefixsingle`.

### 7.4 Arbitrary code around line number

`\Xendbhooklinenumber` `\Xendbhooklinenumber[⟨s⟩]{⟨code⟩}` is used to execute code before line number in endnotes. The code is executed before the `\Xendbeforelinenumber` space and before the `\Xendnotenumfont` font setting.

<code>\Xendahooklinenumber</code>	<code>\Xendahooklinenumber[\langle s \rangle]{\langle code \rangle}</code> is used to execute code after line number in endnotes. The code is executed after the <code>\Xendafternumber</code> space.
<code>\Xendbhookinplaceofnumber</code>	<code>\Xendbhookinplaceofnumber[\langle s \rangle]{\langle code \rangle}</code> is used to execute code before space or symbol which replace line number in endnotes. The code is executed before the <code>\Xendbeforesymmlinenumber</code> space and before the <code>\Xendnotenumfont</code> font setting.
<code>\Xendahookinplaceofnumber</code>	<code>\Xendahookinplaceofnumber[\langle s \rangle]{\langle code \rangle}</code> is used to execute code after space or symbol which replace line number in endnotes. The code is executed after the <code>\Xendaftersymmlinenumber</code> space.

## 7.5 Separator between the lemma and the note

### 7.5.1 For footnotes

<code>\Xlemmaseparator</code>	By default, in a footnote, the separator between the lemma and the note is a right bracket ( <code>\rbracket</code> ) <sup>18</sup> . You can use <code>\Xlemmaseparator[\langle s \rangle]{\langle Xlemmaseparator \rangle}</code> 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 <b>breakable</b> space between the separator and the following text.
<code>\Xbeforelemmaseparator</code>	Using <code>\Xbeforelemmaseparator[\langle s \rangle]{\langle l \rangle}</code> 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.
<code>\Xafterlemmaseparator</code>	Using <code>\Xafterlemmaseparator[\langle s \rangle]{\langle l \rangle}</code> 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.
<code>\Xnolemmaseparator</code>	You can suppress the lemma separator, using <code>\Xnolemmaseparator[\langle s \rangle]</code> , which is simply an alias of <code>\Xlemmaseparator[\langle s \rangle]{}</code> .
<code>\Xinplaceoflemmaseparator</code>	With <code>\Xinplaceoflemmaseparator[\langle s \rangle]{\langle l \rangle}</code> you can add a space if no lemma separator is printed. The default value is 1 em.

### 7.5.2 For endnotes

<code>\Xendlemmaseparator</code>	By default, there is no separator inside endnotes between the lemma and the content of the note. You can use <code>\Xendlemmaseparator[\langle s \rangle]{\langle Xendlemmaseparator \rangle}</code> to change this. The optional argument can be used to specify the series in which it is used. A common value of <code>\Xendlemmaseparator</code> is <code>\rbracket</code> . Note that there is a non-breakable space between the lemma and the separator, but a <b>breakable</b> space between the separator and the following text.
<code>\Xendbeforelemmaseparator</code>	Using <code>\Xendbeforelemmaseparator[\langle s \rangle]{\langle l \rangle}</code> 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.
<code>\Xendafterlemmaseparator</code>	Using <code>\Xendafterlemmaseparator[\langle s \rangle]{\langle l \rangle}</code> 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.
<code>\Xendinplaceoflemmaseparator</code>	With <code>\Xendinplaceoflemmaseparator[\langle s \rangle]{\langle l \rangle}</code> you can add some space if you chose to remove the lemma separator. The default value is 0.5 em.

<sup>18</sup>For polyglossia, when the lemma is RTL, the bracket automatically switches to a left bracket.

## 7.6 Font style

### 7.6.1 For line number

<code>\Xnotenumfont</code>	<code>\Xnotenumfont[⟨s⟩]{⟨command⟩}</code> is used to change the font style for line numbers in critical footnotes ; <code>⟨command⟩</code> must be one (or more) switching command, like <code>\bfseries</code> .
<code>\Xendnotenumfont</code>	<code>\Xendnotenumfont[⟨s⟩]{⟨command⟩}</code> is used to change the font style for line numbers in critical footnotes. <code>⟨command⟩</code> must be one (or more) switching command, like <code>\bfseries</code> .
<code>\notenumfontX</code>	<code>\notenumfontX[⟨s⟩]{⟨command⟩}</code> is used to change the font style for note numbers in familiar footnotes. <code>⟨command⟩</code> must be one (or more) switching command, like <code>\bfseries</code> .

### 7.6.2 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[⟨s⟩]</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[⟨s⟩]</code> to disable it for a specific series.
<code>\Xlemmafont</code> <code>\Xendlemmafont</code>	Use <code>\Xlemmafont[⟨s⟩]{⟨cmd⟩}</code> to apply a $\TeX$ font command to the lemma. For example, to have boldface lemma:  <code>\Xlemmafont{\bfseries}</code>  <code>\Xendlemmafont[⟨s⟩]{⟨cmd⟩}</code> is the same for endnotes.

### 7.6.3 For all notes

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

## 7.7 Wrapping notes

### 7.7.1 Wrapping lemmas

<code>\Xwraplemma</code>	<code>\Xwraplemma[⟨s⟩]{⟨cmd⟩}</code> is used to wrap, in the footnote, the lemma in a $\TeX$ com-
--------------------------	---

mand. For example, with the `bidir` package, to ensure having a lemma written right to left, use `\Xwraplemma{\RL}`.

`\Xwrapendlemma`      `\Xendwraplemma[\langle s \rangle]{\langle cmd \rangle}` is the same for endnotes.

### 7.7.2 Wrapping contents

`\Xwrapcontent`      `\Xwrapcontent[\langle s \rangle]{\langle cmd \rangle}` is used to wrap the footnote contents — excluding the lemma — in a  $\TeX$  command.

For example, if the language of your note is not the same as the language of the lemma, use `\Xwrapcontent{\foreignlanguage{\langle language \rangle}}` (with `babel`) or `\Xwrapcontent{\text{\langle language \rangle}}` (for `babel`).

`\Xendwrapcontent`      `\Xendwrapcontent[\langle s \rangle]{\langle cmd \rangle}` is the same for endnotes.

`\wrapcontentX`      `\wrapcontentX[\langle s \rangle]{\langle cmd \rangle}` is the same for critical footnotes.

## 7.8 Indent of notes content

`\Xparindent`      By default, `reledmac` does not add indentation before the paragraphs inside critical footnotes. Use `\Xparindent[\langle s \rangle]` to enable indentation.

`\parindentX`      By default, `reledmac` does not add indentation before the paragraphs inside familiar footnotes. Use `\parindentX[\langle s \rangle]` to enable indentation.

`\Xhangindent`      For critical notes NOT paragraphed you can define an indent with `\Xhangindent[\langle s \rangle]{\langle l \rangle}`, 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.

`\hangindentX`      For familiar notes NOT paragraphed you can define an indentation with `\hangindentX[\langle s \rangle]{\langle l \rangle}`, 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.

`\Xendhangindent`      For critical endnotes NOT paragraphed you can define an indentation with `\Xendhangindent[\langle s \rangle]{\langle l \rangle}`, 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.

## 7.9 Arbitrary code at the beginning of notes

The three next commands add 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}.}}
```

`\Xbhooknote`      `\Xbhooknote[\langle s \rangle]{\langle code \rangle}` is to be used at the beginning of the critical footnotes.

`\bhooknoteX`      `\bhooknoteX[\langle s \rangle]{\langle code \rangle}` is to be used at the beginning of the familiar footnotes.

`\Xendbhooknote`      `\Xendbhooknote[\langle s \rangle]{\langle code \rangle}` is to be used at the beginning of the endnotes.

## 7.10 Arbitrary code before inserting note

`\Xbeforeinserting`      `\Xbeforeinserting[\langle s \rangle]{\langle code \rangle}` and `\beforeinsertingX[\langle s \rangle]{\langle code \rangle}` are very technical commands.

`beforeinsertingX`

They allow one to add any arbitrary code just before the footnotes are added in the list of footnotes. The main use is to insert text direction code. For example, if you edit right-to-left text with `bidl`, but want your critical footnote be left-to-right, use `\Xbeforeinserting\LTR`. You should also use `\Xwraplemma` to ensure your lemmas are right-to-left in a left-to-right paragraph (7.7.1 p. 44)).

Note that the changes are local to the footnote.

## 7.11 Options for footnotes in columns

### 7.11.1 Alignment

`\Xcolalign` By default, text in footnotes of two or three columns are flush left and 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 `reledmac` setting.

`\RaggedRight` to have text left aligned *with hyphenation* (requires `ragged2e`).

`\raggedleft` to have text right aligned, but *without hyphenation*.

`\RaggedLeft` to have text right aligned *with hyphenation* (requires `ragged2e`).

`\centering` to have text centered, but *without hyphenation*.

`\Centering` to have text centered *with hyphenation* (requires `ragged2e`).

### 7.11.2 Size of the columns

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

`\Xhsizetwocol` `\Xhsizetwocol[⟨s⟩]{⟨l⟩}` is used to change width of a column when critical notes are displaying in two columns. Default value is `.45 \hsizel`.

`\Xhsizethreecol` `\Xhsizethreecol[⟨s⟩]{⟨l⟩}` is used to change width of a column when critical notes are displaying in three columns. Default value is `.3 \hsizel`.

`\hsizetwocolX` `\hsizetwocolX[⟨s⟩]{⟨l⟩}` is used to change width of a column when familiar notes are displaying in two columns. Default value is `.45 \hsizel`.

`\hsizethreecolX` `\hsizethreecolX[⟨s⟩]{⟨l⟩}` is used to change width of a column when familiar notes are displaying in three columns. Default value is `.3 \hsizel`.

## 7.12 Options for paragraphed footnotes and notes grouped by line

### 7.12.1 Mark separation of notes

`\Xafternote` You can add some horizontal space after a note by using `\Xafternote[⟨s⟩]{⟨l⟩}` (for  
`\afternoteX`

critical footnotes) or `\afternoteX[⟨s⟩]{⟨l⟩}` (for familiar footnotes). The default value is 1em plus .4em minus .4em.

`\Xparafootsep` For paragraphed footnotes (see below), you can choose the separator between each  
`\parafootsepX` note by using `\Xparafootsep[⟨s⟩]{⟨text⟩}` for critical notes and `\parafootsepX` for familiar notes. A common separator is the double pipe (`||`), which you can set by using `\Xparafootsep{$\parallel$}`.

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.

### 7.12.2 Ragged text

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

`\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.

## 7.13 Options for block of notes

### 7.13.1 Grouping notes by line

`Xgroupbyline` If you do not use `\Xarrangement{paragraph}`, you may want to group all the critical footnotes related to the same line in the same paragraph. In this case, use `\Xgroupbyline[⟨series⟩]`.

In many cases, you might like to use it in combination with `\Xnumberonlyfirstinline` (7.2.1 p. 38).

`Xgroupbylineseparetwolines` Note that the `\Xafternote` and `\Xparafootsep` settings are used to determine space and content between footnotes (7.12 p. 46). Suppose you have two notes on line 1 which overlap lines 1 and 2. This last note will be printed, if you use `\Xgroupbyline` in the same group as the previous one. In the case you want that note to be distinct, you must use both `\Xgroupbyline` and `\Xgroupbylineseparetwolines[⟨s⟩]`.

In many cases, you might like to use it in combination with `\Xnumberonlyfirstintwolines` (7.2.1 p. 38).

### 7.13.2 Text before notes

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

### 7.13.3 Code before notes

`\Xhookgroup` While `\Xtxtbeforenotes` is for typesetting code before notes, `\Xhookgroup` and  
`\hookgroupX` (respectively for critical and familiar) are for executing code before a groups of notes, between the rules and the printing of the notes.

### 7.13.4 Spacing

`\Xbeforenotes` You can change the vertical space 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 L<sup>A</sup>T<sub>E</sub>X footnote rule used by reledmac 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 L<sup>A</sup>T<sub>E</sub>X footnote rule, which is used by reledmac, decreases 3pt. These 3pt are not changed by this command.**

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

`\prenotesX` You can set the space before the first printed (in a page) series of familiar notes to be different from the space before other series. The default value is 0pt. You can do this with `\prenotesX{⟨l⟩}`. You can disable this feature by setting the length to 0pt.

### 7.13.5 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 L<sup>A</sup>T<sub>E</sub>X footnote rule, which is used by reledmac, adds 2.6pt. These 2.6pt are not changed by this command.**

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

**Be careful, the standard L<sup>A</sup>T<sub>E</sub>X footnote rule, which is used by reledmac, adds 2.6pt. These 2.6pt are not changed by this command.**

### 7.13.6 Maximum height

`\Xmaxhnotes` By default, one series of critical notes can take up to 80% of `\vsize`, before being broken to the next page. If you want to change the size use `\Xmaxhnotes[⟨s⟩]{⟨l⟩}`. 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[⟨s⟩]{⟨l⟩}` is the same as previous, but for familiar footnotes.  
Note that in many cases, you should call these commands in the begin of the document, because the `\vsize` in the preamble is not the same as `\vsize` after the preamble. That why we recommend to you to add in your preamble

```
\AtBeginDocument{
  \maxhnotesX{0.8\textheight}
  \Xmaxhnotes{0.8\textheight}
}
```

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



### 7.13.7 Width

`\Xwidth` `\Xwidth[⟨s⟩]{⟨l⟩}` sets the total width of critical footnotes. `\widthX[⟨s⟩]{⟨l⟩}` does the same for familiar footnotes.

⟨l⟩ can be a length expression, parsable with `\dimexpr`. For example:

```
\Xwidth{\columnwidth+\marginparsep+\ledrsnotewidth}
\widthX{\columnwidth+\marginparsep+\ledrsnotewidth}
```

Note that changes the width of the block of notes. If you want to change the width of each column when typesetting notes in columns, use `\Xhsizetwocol`, `\Xhsizethreecol`, `\hsizetwocolX`, `\hsizethreecolX`, see 7.11.2 p. 46.

## 7.14 Footnotes and the `reledpar` columns

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

`\noteswidthliketwocolumnsX`

- `\Xnoteswidthliketwocolumns[⟨s⟩]` to create critical notes with a two-column size width.
- `\noteswidthliketwocolumnsX[⟨s⟩]` to create familiar notes with a two-column size width.

## 7.15 Endnotes in one paragraph

`\Xendparagraph` By default, any new endnote starts a new paragraph. Use `\Xendparagraph[⟨s⟩]` 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[⟨s⟩]{⟨l⟩}`. The default value is `1em plus .4em minus .4em`.

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

# 8 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.

`\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.

`\select@lemmafонт` We will briefly discuss `\select@lemmafонт` 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.

## 9 Verse

### 9.1 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.

If you need to add brackets directly after `\stanza`, `&` or `\&`, add `\norelax`, as in other case the brackets will be interpreted as delimitation for optional argument (cf. 9.8 p. 52)

### 9.2 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 in 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. 9.6 p. 52.

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.

### 9.3 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 indicate that they are repeated, defining the value of the `stanzaindentrepetition` counter at  $n$ . For example:

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

is like

```
\setstanzaindent{5,1,0,1,0,1,0,1,0,1,0}
```

**Be careful: the feature is changed in eledmac 1.5.1. See Appendix A.3 p. 360.**

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  $\text{\TeX}$ '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.

### 9.4 Manual stanza indent

```
\stanzaindent
\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`.

### 9.5 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.

## 9.6 Hanging symbol

It is possible to insert a symbol in each line of hanging verse, as in French typography; for example, the opening bracket ‘[’. To insert it in `reledmac`, use macro `\sethangingsymbol{<h>}` with this code. In the example of French typography, do

`\sethangingsymbol`

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

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

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

## 9.7 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 18.2 p. 69 for further details.

## 9.8 Content before/after verses

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

- The `\stanza` command can take an optional argument (in brackets). Its content will be printed before the stanza. A `\noindent` is inserted before the content of first optional argument. If you don’t want this `\noindent`, you can use the second optional argument (also in brackets):

```
\stanza[foo] % \noindent is inserted before foo.
\stanza[][foo] % There is no \noindent inserted before foo.
```

<code>\AtEveryStanza</code>	<ul style="list-style-type: none"> <li>• Use <code>\AtEveryStanza{&lt;arg&gt;}</code> to automatically add content before the stanza (not in the same paragraph).</li> </ul> <p>Note that a <code>\noindent</code> will be inserted before the argument, and, consequently, a <code>\parskip</code>. You can use the starred version of <code>\AtEveryStanza</code> to avoid this <code>\noindent</code>.</p>
<code>\AtStartEveryStanza</code>	<ul style="list-style-type: none"> <li>• Use <code>\AtStartEveryStanza</code> to automatically add content at the beginning of stanza (in the same paragraph).</li> <li>• <code>&amp;</code> can be replaced by <code>\newverse</code> with two optional arguments (in brackets). The first will be printed after the current verse, the second before the next verse.</li> </ul> <p>A <code>\noindent</code> is automatically inserted before the contents of these optional arguments.</p> <p>Use a third and fourth optional argument to not add these <code>\noindents</code> (to add content respectively after the current verse / before the next verse).</p> <ul style="list-style-type: none"> <li>• Use <code>\AtEveryPend{&lt;arg&gt;}</code> to automatically add content after verses (including the final one) and <code>\AtEveryPstart{&lt;arg&gt;}</code> to automatically add content before verses (including the first one).</li> <li>• <code>\&amp;</code> can take an optional argument (in brackets). Its content will be printed after the stanza.</li> </ul>
<code>\AtEveryStopStanza</code>	<ul style="list-style-type: none"> <li>• Use <code>\AtEveryStopStanza</code> to automatically add content after the end of stanzas (not in the same paragraph).</li> </ul> <p>Note that a <code>\noindent</code> will be inserted before the argument, and, consequently, a <code>\parskip</code>. You can use the starred version of <code>\AtEveryStopStanza</code> to avoid this <code>\noindent</code>.</p>
<code>\AtStartEveryStanza</code>	<ul style="list-style-type: none"> <li>• Use <code>\AtStartEveryStanza</code> to automatically add content at the end of stanza (in the same paragraph).</li> </ul>

## 9.9 Numbering stanza

<code>\numberstanzatrue</code>	<p>If you want to automatically number stanzas, use <code>\numberstanzatrue</code>. In this case, the line number will restart at each <code>\stanza</code>.</p> <p>If you want to disable this feature again, use <code>\numberstanzafalse</code>.</p> <p>You can use this feature in combination with <code>\Xstanza</code> (7.2.8 p. 40).</p>
<code>\numberstanzafalse</code>	
<code>thestanza</code>	<p>You can redefine <code>\thestanza</code> to change the aspect of stanza number. Default value is:</p> <pre> \renewcommand{\thestanza}{%   \textbf{\arabic{stanza}}% }</pre>

You can change the value of the `stanza` counter with the usual commands of  $\TeX$ .  
`\stanzanumwrapper` You can redefine `\stanzanumwrapper` in order to modify the way the stanza number is inserted in the flow of text. Default value is:

```
\newcommand{\stanzanumwrapper}[1]{%
  \flagstanza{#1}%
}
```

## 9.10 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`.

## 9.11 Notes on empty lines

Since v2.3.0 of `reledmac`, empty lines when typesetting verses no longer produce new paragraphs, and consequently, do not insert vertical spaces. Use optional argument of `\stanza` or `\newverse` to insert vertical space (9.8 p. 52).

# 10 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 6.5) 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}[⟨pos⟩]{⟨width⟩}`.

The required `⟨width⟩` argument is the text width for the environment. The optional `⟨pos⟩` 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}`

## 11 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.

### 11.1 Basic use

`\edlabel` First you place a label in the text using the command `\edlabel{<lab>}`. `<lab>` can be almost anything you like, including letters, numbers, punctuation, or a combination—anything but spaces; you might type `\edlabel{toves-3}`, for example.<sup>19</sup>

`\edpageref` Elsewhere in the text, either before or after the `\edlabel`, you can refer to its

`\edlineref` location with `\edpageref{<lab>}`, `\edlineref{<lab>}`, `\edsublineref{<lab>}` or

`\sublineref` `\pstartref{<lab>}`, that will produce, respectively, the page, line, sub-line and pstart

`\pstartref` on which the `\edlabel{<lab>}` command occurred.

Note that the `\edlineref` command insert the side flag after the line number.

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

The `\edlabel` command works by writing macros to `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 use `\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.)

### 11.2 Cross-referencing to a critical note

If you want to refer to a word which is a lemma word, the `\edlabel` command should be in the first argument of `\edtext` command.

If you want to refer to the content of a `\Xfootnote`, the line and subline number printed will be the start line.

If you want to refer to starting and ending lines, you should use `\appref` and related tools (11.6.2 p. 57).

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

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

`\xlineref`

`\xsublineref`

`\xpstartref`

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 to use the reference in a context where  $\TeX$  is looking for a number, such a warning will lead to a complaint that the number is missing. This is the case for references used within the argument to `\linenum`, for example (see 6.2.5 p. 27).

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.)
- With `reledpar`, the `\xlineref` does not insert the right side flag, in order to obtain a line number. Use `\xflagref` to obtain the side flag, depending of your flag.

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

`\xxref`

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 automatically calls `\linenum` (q.v., 6.2.5 p. 27 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.

For example, one might use the following:

```
\beginnumbering

\pstart
\edlabel{Queritur}Queritur utrum metaphysica sit scientia una.
\pend

\pstart
\edtext{Et videtur quod non\edlabel{non}.}{\xxref{Queritur}{non}\lemma{queritur \dots} non}
\pend

\endnumbering
```

---

<sup>19</sup>More precisely, you should stick to characters in the  $\TeX$  categories of “letter” and “other”.



## 11.4 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 type `\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.

## 11.5 Normal $\text{\LaTeX}$ cross-referencing

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

## 11.6 References to start and end lines

### 11.6.1 Reference to main text lines

Many times, you may want to make a cross-reference to a passage that is defined by a start line and an end line. `reledmac` provides specific tools for this scenario.

`\edlabelS` Use `\edlabelS{<label>}` to mark the start line of the passage.

`\edlabelE` Use `\edlabelE{<label>}` to mark the end the end line of the passage. These two commands just create to label which are named `<label>:start` and `<label>:end`.

`\edlabelSE` Use `\edlabelSE{<label>}` to mark just one location in the text. Contrary to a classical `\edlabel`, the `<label>` could be use with `\Seref` and `\Serefwithpage`.

`\Seref` The main utility is to use them with three other commands. `\Seref{<label>}` will make a cross-reference printed as a reference in critical footnotes.

`\Serefwithpage` `\Serefwithpage` will make a cross-reference printed as a reference in critical endnotes.

`\Serefonlypage` `\Serefonlypage` will make a cross-reference printed only with page number.

### 11.6.2 References to lines that are commented on 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` (6.2.5 p. 27) 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 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.  
`\apprefwithpage`

### 11.6.3 Settings

`\setapprefprefixsingle`  
`\setapprefprefixmore`

**Specific to these tools** If you use `\apprefprefixsingle{⟨prefix⟩}`, `⟨prefix⟩` will be printed before the line numbers of a `\appref`-reference. If you use `\apprefprefixmore{⟨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 do not use `\setapprefprefixmore`, the argument of `\setapprefprefixsingle` will be used in any case.

`\setSerefprefixsingle`  
`\setSerefprefixmore`

`\setSerefprefixsingle` and `\setSerefprefixmore` are similar for `\Seref` command.

`\setSerefonlypageprefixsingle`  
`\setSerefonlypageprefixmore`

Use `\setSerefonlypageprefixsingle{⟨prefix⟩}` to set the page prefix for `\Serefonlypage` when there is only one page. Use `\setSerefonlypageprefixmore{⟨prefix⟩}` to set it when there is more than one page. For example:

```
\setSerefonlypageprefixsingle{p.~}
\setSerefonlypageprefixmore{pp.~}
```

Note that if you do not use `\setSerefonlypageprefixmore`, the value of `\setSerefonlypageprefixsingle` is used instead.

**Also note that `\setSerefonlypageprefixsingle` is only a shortcut for `\Xendbeforepagenumber` (see 11.6.3 p. 58). So if you use `\Xendbeforepagenumber` without any optional argument, it will override this setting.**

**Linked to setting of critical endnotes and footnotes** Some commands who set the appearance of line numbers in critical footnotes also set the appearance of line numbers in `\appref` and `\Seref` if you call them *without the optional series argument*.

These commandes are the following:

- `\Xlineflag` (for `reledpar`), enabled by default.
- `\Xlinerangeseparator`
- `\Xmorethantwolines`
- `\Xsublinesep`
- `\Xtwolines`
- `\Xtwolinesbutnotmore`
- `\Xtwolinesonlyinsamepage`

If you want to make settings specific to `\appref` or `\Seref`, just call them with an optional argument containing a comma-separated list of command names (for example `appref,Seref`) or with a suffix equal to the command name (for example `appref`).

The same principle is available for `\apprefwithpage`, `\Serefwithpage` and `\Serefonlypage` with the following commands:

- `\Xendafterpagenumber` (not for `\Serefonlypage`)
- `\Xendbeforepagenumber`
- `\Xendlineflag` (for `reledpar`), enabled by default.
- `\Xendlineprefixmore`
- `\Xendlineprefixsingle`
- `\Xendlinerangeseparator`
- `\Xendmorethantwolines`
- `\Xendsublinesep`
- `\Xendtwolines`
- `\Xendtwolinesbutnotmore`
- `\Xendtwolinesonlyinsamepage`

**For one specific command** When calling `\appref` and `\Seref`, you can use as a first optional argument, in brackets (`[]`), any optional argument which can be used for critical footnotes (6.2.2 p. 25).

When calling `\apprefwithpage`, `\Serefwithpage` or `\Serefonlypage` you can use as a first optional argument, in brackets (`[]`), any optional argument which can be used for critical endnotes (6.2.3 p. 25).

## 11.7 Compatibility with *xr* package

The `\externaldocument` command of the `\xr` package allows making cross-references from an external document, with the standard  $\TeX$  commands `\label` and `\ref` (and related).

To use it with the `reledmac` cross-reference commands (i.e. `\edlabel` and related), you must do the following:

1. Load the `xr` package.
2. Load the `reledmac` package.
3. Use the `\externaldocument` document command.

## 12 Side notes

### 12.1 Basics

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

`\ledinnernote`      `\ledinnernote{<text>}` will put `<text>` into the inner margin level with where the  
`\ledouternote`

command was issued. Similarly, `\ledouternote{⟨text⟩}` puts `⟨text⟩` in the outer margin.

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

to typeset `\ledsidenotes` in the right hand margin. This is the opposite of 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 note commands for the same side are called in the same line, they will be appended and separated by a comma.

**The notes will appear only after the second  $\LaTeX$  run. If the note positions change in your .tex file, you need two runs to get the correction position in the output file. You are strongly encouraged to use tools like *latexmk*, to be sure to get the correct number of runs.**

## 12.2 Setting

### 12.2.1 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`.

### 12.2.2 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).

### 12.2.3 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`.

### 12.2.4 Font

`\ledlsnotefontsetup`    These macros specify how the sidenote texts are to be typeset. The initial definitions are:  
`\ledrsnotefontsetup`    `\newcommand*{\ledlsnotefontsetup}{\raggedleft\footnotesize}% left`  
                              `\newcommand*{\ledrsnotefontsetup}{\raggedright\footnotesize}% right`

These can of course be changed to suit.

### 12.2.5 Separator between notes

`\setsidenotessep` 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 `\setsidenotessep{<sep>}`.

## 13 Indexing

### 13.1 Basics

`\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, because of the internal indexing mechanism of `reledmac`. That means you must first run (Xe/Lua) $\TeX$  three times, then run `makeindex`, and then finally run (Xe/Lua) $\TeX$  again, in order to get an index with the right page numbers.

If the `imakeidx` or `indextools` package is used then the macro takes an optional argument, 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 `reledmac`.
3. Declare the index with the macro `\makeindex` of `imakeidx` and `indextools`.

Also note that using `\edtext` in familiar footnotes refers to the line where the footnotes are called

### 13.2 Referring to critical notes

If you want to refer to a word inside an `\edtext{<lemma>}{<app>}` command, `\edindex` should be defined inside the first argument, e.g.,

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

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

```
\newcommand{\ledinnotemark}[1]{#1\emph{n}}
```

### 13.3 Separator between page and line numbers

`\pagelinesep` The page & linenum 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.

You can reconfigure it, this example defines a colon as separator:

```
\renewcommand{\pagelinesep}{:}
```

However, you also have to configure your `.ist` index style file. For example if you use `:` as separator<sup>20</sup>.

```
page_compositor ":"
```

Read the MAKEINDEX program’s handbook about the `.ist` file.

### 13.4 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-linenum`

An example of such a file is provided in the “examples” folder. Read the `xindy` handbook to learn how to use it.<sup>21</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:

1. Use `xindy+hyperref` option when loading the `reledmac` package. When you run  $(\text{Xe/Lua})\text{\LaTeX}$  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`.

<sup>20</sup>For further detail, you can read <http://tex.stackexchange.com/a/32783/7712>.

<sup>21</sup>Or, for people who read French, read <http://geekographie.maieul.net/174>.

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>22</sup>.

## 13.5 Advanced setting

`\edindexlab` The `\edindex` process uses a `\label` and `\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.

## 14 Glossary

`reledmac` provides mechanism to make glossaries with the `glossaries` package, referring not to the page, but to the page and line.

### 14.1 Preamble setting

The standard compositor between page and line number in `reledmac` is a dash, while `glossaries` uses, by default, a dot. Consequently, you must:

- Or set `.glossaries`:  

```
\glsSetCompositor{-}
```
- Or set `reledmac`:  

```
\renewcommand{\pagelinesep}{-}
```

In this case, the above will have consequences for your use of `\edindex` and you should set your `.ist` file (13.3 p. 62).

### 14.2 Commands

The `\gls`, `\Gls`, and related commands of `glossaries` packages have a prefixed version with `ed`, which refers to the page line. The argument are the same as for the standard commands. So for example:

```
\edgls[<options>]{<label>}[<insert>]
```

## 15 Tabular material

$\text{\LaTeX}$ ’s normal `tabular` and `array` environments cannot be used where line numbering is being done; more precisely, they can be used but with odd results, so don’t use

---

<sup>22</sup>These are the recommended lines to provide the best possible compatibility between `hyperref` and `xindy`, even without using `reledmac`.







`\edatleft`      `\edatleft[ $\langle math \rangle\{\langle symbol \rangle\}\langle halfheight \rangle$ ]` typesets the math  $\langle symbol \rangle$  as  $\left\{\langle symbol \rangle\right\}$  with the optional  $\langle math \rangle$  centered before it. The  $\langle symbol \rangle$  is twice  $\langle halfheight \rangle$  tall. The `\edatright` macro is similar and it typesets  $\right\{\langle symbol \rangle\}$  with  $\langle math \rangle$  centered after it.

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

$$left = \left( \begin{array}{ccc} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{array} \right) = right$$

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

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}
```

	$A$ 1   2   3 $B$ 1   3   6 $C$ 1   4   8 $D$ 1   5   0	After
Before		

`\edvertline`      The macro `\edvertline{ $\langle height \rangle$ }` draws a vertical line  $\langle height \rangle$  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}
```

<i>a</i>	<i>b</i>	<i>C</i>	<i>d</i>	
<i>v</i>	<i>w</i>	<i>x</i>	<i>y</i>	
<i>m</i>	<i>n</i>	<i>o</i>	<i>p</i>	
<i>k</i>		<i>L</i>	<i>cvb</i>	

The `\edvertdots` macro is similar to `\edvertline` except that it produces a vertical dotted instead of a solid line.

## 16 Sectioning commands

### 16.1 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` (5.2.3 p. 18):

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

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

### 16.2 Sectioning commands with line numbering and critical notes

You have to use the following commands:

- `\eledchapter[\langle text \rangle]{\langle critical text \rangle}`,
- `\eledchapter*`,
- `\eledsection[\langle text \rangle]{\langle critical text \rangle}`,
- `\eledsection*`,
- `\eledsubsection[\langle text \rangle]{\langle critical text \rangle}`,
- `\eledsubsection*`,
- `\eledsubsubsection[\langle text \rangle]{\langle critical text \rangle}`,
- `\eledsubsubsection*`.

These 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

```

After the first run, you will see only the text. This is normal. After the second run, you will see the formatting. Finally, with the third run, you will see the table of contents.

For technical reasons, the page break before `\elechapter` cannot be added automatically. You have to insert it manually via `\beforeeledchapter`, which must be called outside of a numbered section.

### 16.3 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` files, save memory, and make `reledmac` run faster.

## 17 Quotation environments

The quotation and quote environments can be used so that the 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 `\pstart ... \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.

## 18 Page breaks

### 18.1 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.**

`\ledpbsetting` These two commands take effect at the beginning of line in which they are called. For example, if you call `\ledpb` at l. 444, then 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, l. 444 will be on p.  $n$  and l. 445 will be on p.  $n + 1$ .

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 be out of sync.

## 18.2 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 and no more. It works on the third run, or on the fourth run if using `reledpar`. By default, when a long verse runs 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.

## 19 Miscellaneous

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

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

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

If you would prefer some other style, you could put something like this in the preamble:

```
\ifledfinal\else
  \renewcommand{\showlemma}[1]{\textbf{#1}}% or simply ...[1]{#1}
\fi
```

## 19.1 Known and suspected limitations

### 19.1.1 Non-standard geometry

If you use classes other than `article` or `book`, or if you use the `geometry` package, you should use `maxhnotesX` and/or `\Xmaxhnotes` as explained in 7.13.6 p. 48 in order to prevent footnotes from overlapping the bottom margin.

### 19.1.2 floatrow package compatibility

The `floatrow` package must be loaded before the `reledmac`.

### 19.1.3 ‘No room for a new’

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

In order to prevent such problems, the first thing is to use the options to optimize `reledmac`. For example, if you need only two series of notes, use the `series={A,B}` option. Read 16.3 p. 68 in order to know which are the available options.

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

‘**no room for a new count**’ is often caused by `biblatex` being used at the same time. Load `reledmac` (and `reledpar`) *before* `biblatex`.

‘**no room for a new write**’ can be caused by 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.

### 19.1.4 Marginal notes

In general, `reledmac`’s system for adding marginal line numbers breaks anything that makes direct use of the  $\LaTeX$  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.

### 19.1.5 Paragraph shape

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

`\ballast`  $\LaTeX$  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, insert `\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.

### 19.1.6 Paragraphed footnotes

The restriction on explicit line-breaking in paragraphed footnotes, mentioned on 7.1 p. 37, and described in more detail on XII.6.3 p. 174, 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.

If you use more than one series of paragraphed notes, it may happen, in some particular cases, that only the footnote rule, with no accompanying footnotes, be printed. In this case use `reledmac` package option `nopenalties` which should solve the problem, but also may produce widow or orphan lines. For the time being, we have no solution of this problem.

`\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}
```

Note that you must call it *before* `\Xarrangement{paragraph}` or `\arrangementX{paragraph}`.

Any settings to ‘geometry’ must be made before `\Xarrangement` / `\arrangementX`.

Finally, in many cases you should use `\Xmaxhnotes` and / or `\maxhnotesX` (7.13.6 p. 48), in order to define the maximum height relative to `\textheight` and not to `\vsize`, because the `\vsize` value is not the same inside and outside of the preamble.

### 19.1.7 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>23</sup> you will find  $\TeX$  whinging ‘Missing { inserted’, and then things start to fall apart. The trick in this case is to specify either:

<sup>23</sup>Reported by Dirk-Jan Dekker in the CTT thread ‘Incompatibility of “color” package’ on 2003/08/28.

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

or

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

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

## 19.2 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



## I Implementation overview

We present the `reledmac` code in roughly the order in which it is used during a run of  $\text{\TeX}$ . 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  $\text{\LaTeX}$  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 XXII) 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  $\text{\LaTeX}$  macros.
- Replace user-level  $\text{\TeX}$  counts by  $\text{\LaTeX}$  counters.
- Use the  $\text{\LaTeX}$  font handling mechanisms.
- Use  $\text{\LaTeX}$  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}[2017/06/04 v2.21.0 typesetting critical editions]
4 %

```

### 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 \RequirePackage{xkeyval}
\ifnoend@ %
\ifnofamiliar@
\ifnoledgroup@ The parledgroup option is for reledpar. However, it has consequence on reledmac
\ifparapparatus@ internal command. So we need to define the boolean now.
\ifnoquotation@ \newif\ifparledgroup
\iflednopbinverse %
\ifparledgroup
\ifwidthliketwocolumns And now, the options of reledmac.
\ifxindy@ \DeclareOptionX{series}[A,B,C,D,E]{\xdef\default@series{#1}}
\ifxindyhyperref@ \ExecuteOptionsX{series}%
\ifeledmaccompat@
12 \newif\if@noeled@sec%
13 \DeclareOptionX{noeledsec}{\@noeled@sectrue}
14
15 \newif\ifnocritical@%
16 \DeclareOptionX{nocritical}{\nocritical@true}%
17
18 \newif\ifnofamiliar@%
19 \DeclareOptionX{nofamiliar}{\nofamiliar@true}%
20
21 \newif\ifnoledgroup@%
22 \DeclareOptionX{noledgroup}{\noledgroup@true}%
23
24 \newif\ifnoend@%
25 \DeclareOptionX{noend}{%
26 \let\l@dend@open\@gobble%
27 \let\l@dend@close\relax%
28 \global\let\l@dend@stuff=\relax%
29 \noend@true%
30 }%
31
32 \newif\ifnoquotation@
33 \DeclareOptionX{noquotation}{\noquotation@true}
34
35 \newif\ifledfinal
36 \DeclareOptionX{final}{\ledfinaltrue}
37 \DeclareOptionX{draft}{\ledfinalfalse}

```

```

38 \ExecuteOptionsX{final}
39
40 \newif\ifparapparatus@
41 \DeclareOptionX{parapparatus}{\parapparatus@true}
42
43 \newif\iflednopbinverse
44 \DeclareOptionX{nopbinverse}{\lednopbinversetrue}
45
46 \newif\ifwidthliketwocolumns%
47 \DeclareOptionX{widthliketwocolumns}{\widthliketwocolumnstrue}%
48
49 \newif\ifcontinuousnumberingwithcolumns
50 \DeclareOptionX{continuousnumberingwithcolumns}{\
continuousnumberingwithcolumnstrue}%
51
52 \newif\ifxindy@
53 \DeclareOptionX{xindy}[eledmac-markup-attr.xdy]{%
54   \AtBeginDocument{\immediate\openout\eledmac@xindy@out=#1}%
55   \newwrite\eledmac@xindy@out%
56   \xindy@true%
57   \gdef\eledmacmarkuplocdepth{:depth 1}%
58   \AtEndDocument{\immediate\closeout\eledmac@xindy@out}%
59 }%
60
61 \newif\ifxindyhyperref@
62 \DeclareOptionX{xindy+hyperref}{%
63   \xindyhyperref@true%
64 }%
65
66 \newif\ifeledmaccompat%
67 \DeclareOptionX{eledmac-compat}{%
68   \eledmaccompat@true%
69 }%
70 \DeclareOptionX{nopenalties}{%
71   \AtBeginDocument{\let\add@penalties\relax}%
72 }
73 \def\l@auxdir{}%
74 \DeclareOptionX{auxdir}{%
75   \xdef\l@auxdir{#1}%
76 }%
77 %

```

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.

```

78 \ProcessOptionsX*\relax
79
80 %

```

## 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  $\text{Lua}\TeX$  or  $\text{Xe}\TeX$  is running, and `ragged2e` to manage ragged justification for paragraphed notes.

```

81 \RequirePackage{xargs}
82 \RequirePackage{etoolbox}
83 \@ifl@t@r\fmtversion{2015/10/01}
84 {\ifboolexpr{not test{\@ifl@t@r\fmtversion{2016/03/31}} or (test{\
85   ifdefstring{\fmtversion}{2016/03/31}} and test {\ifnumless{\patch@level
86   }{3}})}}%
87   {\PackageWarning{reledmac}{You are using a LaTeX version older than
88   2016/03/31 patch 3.%
89   \MessageBreak You are strongly encouraged to use a newer version.}}}%
90   }%
91 {\RequirePackage{etex}}%
92 \csname reserveinserts\endcsname{32}%
93 }%
94 \RequirePackage{suffix}
95 \RequirePackage{xstring}
96 \RequirePackage{ifluatex}
97 \RequirePackage{ragged2e}
98 \RequirePackage{ifxetex}%
99 %

```

## II.5 Compatibility with $\text{Lua}\TeX$

Here, we enable some primitives for  $\text{Lua}\TeX$ .

```

98 \ifx\directlua\undefined\else%
99   \directlua{tex.enableprimitives("",{"texdir","pardir","bodydir"})}
100 \fi
101 %

```

## II.6 Boolean flags

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

```

102 \newif\ifl@dmemoir
103 \@ifclassloaded{memoir}{\l@dmemoirtrue}{\l@dmemoirfalse}
104
105 %

```

`\if@ledgroup` Flag set to true inside a `ledgroup` environment.

```

106 \newif\if@ledgroup%
107 %

```

**\ifl@imakeidx** Define a flag for if the imakeidx package has been used.

```

108 \newif\ifl@imakeidx
109 \@ifpackageloaded{imakeidx}{\l@imakeidxtrue}{}%False is the default value
110 %

```

**\ifl@indextools** Define a flag for if the indextools package has been used.

```

111 \newif\ifl@indextools%
112 \@ifpackageloaded{indextools}{%
113   \l@indextoolstrue%
114   \l@imakeidxtrue%
115   \let\imki@wrindexentry\indtl@wrindexentry%
116 }{}%
117 %

```

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`.

**\ifl@footmisc** Define a flag if the footmisc package has been loaded.

```

118 \newif\ifl@footmisc
119 \@ifpackageloaded{footmisc}{\l@footmisctrue}{}%False is the default value
120 %

```

**\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.

```

121 \ifdefined\if@RTL{\newif\if@RTL}
122 %

```

**\if@firstlineofpage** `\if@firstlineofpage` is set to TRUE at the first line of every page. `\if@firstlineofpageR` is for the right side.

```

123 \newif\if@firstlineofpage%
124 \newif\if@firstlineofpageR%
125 %

```

## II.7 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.

```

126 \newcommand{\reledmac@warning}[1]{\PackageWarning{reledmac}{#1}}
127 %

```

`\reledmac@error` Write an error message.

```
128 \newcommand{\reledmac@error}[2]{\PackageError{reledmac}{#1}{#2}}
129 %
```

```
\led@err@NumberingStarted30 \newcommand*{\led@err@NumberingStarted}{%
\led@err@NumberingNotStarted31 \reledmac@error{Numbering has already been started}{\@ehc}}
\led@err@NumberingShouldHaveStarted32 \newcommand*{\led@err@NumberingNotStarted}{%
133 \reledmac@error{Numbering was not started}{\@ehc}}
134 \newcommand*{\led@err@NumberingShouldHaveStarted}{%
135 \reledmac@error{Numbering should already have been started}{\@ehc}}
136 %
```

```
\led@err@edtextoutsidestart37 \newcommand*{\led@err@edtextoutsidestart}{%
138 \reledmac@error{\string\edtext\space outside numbered paragraph (\...pstart
\pend)}{\@ehc}}%
139 %
```

```
\led@mess@NotesChanged40 \newcommand*{\led@mess@NotesChanged}{%
141 \typeout{reledmac reminder: }%
142 \typeout{ The number of the footnotes in this section
143 has changed since the last run.}%
144 \typeout{ You will need to run LaTeX two more times
145 before the footnote placement}%
146 \typeout{ and line numbering in this section are
147 correct.}}
148 %
```

```
\led@mess@SectionContinued49 \newcommand*{\led@mess@SectionContinued}[1]{%
150 \message{Section #1 (continuing the previous section)}}
151 %
```

```
\led@err@LineationInNumbered52 \newcommand*{\led@err@LineationInNumbered}{%
153 \reledmac@error{You can't use \string\lineation\space within
154 a numbered section}{\@ehc}}
155 %
```

```
\led@warn@BadLineation56 \newcommand*{\led@warn@BadLineation}{%
\led@warn@BadLinenummargin57 \reledmac@warning{Bad \string\lineation\space argument}}
\led@warn@BadLockdisp58 \newcommand*{\led@warn@BadLinenummargin}{%
\led@warn@BadSublockdisp59 \reledmac@warning{Bad \string\linenummargin\space argument}}
160 \newcommand*{\led@warn@BadLockdisp}{%
161 \reledmac@warning{Bad \string\lockdisp\space argument}}
162 \newcommand*{\led@warn@BadSublockdisp}{%
163 \reledmac@warning{Bad \string\sublockdisp\space argument}}
164 %
```

```

\led@warn@NoFile65 \newcommand*{\led@warn@NoFile}[1]{%
166 \reledmac@warning{File `#1' not found}}
167 %

\led@warn@LineFileObsolete68 \newcommand*{\led@warn@Obsolete}[1]{%
169 \reledmac@warning{Line-list file #1 was obsolete. We have not read it.
Please run LaTeX again.}}
170 %

\led@warn@BadAdvancelineSubline71 \newcommand*{\led@warn@BadAdvancelineSubline}{%
\led@warn@BadAdvancelineLine72 \reledmac@warning{\string\advanceline\space produced a sub-line
173 number less than zero.}}
174 \newcommand*{\led@warn@BadAdvancelineLine}{%
175 \reledmac@warning{\string\advanceline\space produced a line
176 number less than zero.}}
177 %

\led@warn@BadSetline78 \newcommand*{\led@warn@BadSetline}{%
\led@warn@BadSetlinenum79 \reledmac@warning{Bad \string\setline\space argument}}
180 \newcommand*{\led@warn@BadSetlinenum}{%
181 \reledmac@warning{Bad \string\setlinenum\space argument}}
182 %

\led@err@PstartNotNumbered83 \newcommand*{\led@err@PstartNotNumbered}{%
\led@err@PstartInPstart84 \reledmac@error{\string\pstart\space must be used within a
\led@err@PendNotNumbered85 numbered section %
\led@err@PendNoPstart86 (\string\...beginnumbering\string\endnumbering)}{\@ehc}}%
\led@err@AutoparNotNumbered87 \newcommand*{\led@err@PstartInPstart}{%
\led@err@NumberingWithoutPstart88 \reledmac@error{\string\pstart\space encountered while another
189 \string\pstart\space was in effect}{\@ehc}}
190 \newcommand*{\led@err@PendNotNumbered}{%
191 \reledmac@error{\string\pend\space must be used within a
192 numbered section}{\@ehc}}
193 \newcommand*{\led@err@PendNoPstart}{%
194 \reledmac@error{\string\pend\space must follow a \string\pstart}{\@ehc}}
195 \newcommand*{\led@err@AutoparNotNumbered}{%
196 \reledmac@error{\string\autopar\space must be used within a
197 numbered section}{\@ehc}}
198 \newcommand*{\led@err@NumberingWithoutPstart}{%
199 \reledmac@error{\string\beginnumbering...\string\endnumbering\space
without \string\pstart}{\@ehc}}%
200 %

```

```

\led@warn@BadAction201 \newcommand*{\led@warn@BadAction}{%
202   \reledmac@warning{Bad action code, value \next@action.}}
203   %

```

```

\led@warn@DuplicateLabel204 \newcommand*{\led@warn@DuplicateLabel}[1]{%
\led@warn@AppLabelOutSecondArgEdtext205   \reledmac@warning{Duplicate definition of label `#1'\@gobble}%
\led@warn@RefUndefined206   \@latex@warning@no@line{Label `#1' multiply defined}%
\led@warn@RefUndefined207 }%
208 \newcommand*{\led@warn@AppLabelOutSecondArgEdtext}[1]{%
209   \reledmac@warning{\string\applabel\space outside of the second argument
of an \string\edtext\space `#1' on page \thepage.}}%
210 \newcommand*{\led@warn@RefUndefined}[1]{%
211   \G@refundefinedtrue%
212   \reledmac@warning{Reference `#1' on page \thepage\space undefined.%
Using `000'.}%
213   \@latex@warning{Reference `#1' undefined\on@line}%
214 }%
215 \newcommand*{\led@warn@pairRefUndefined}[1]{%
216   \G@refundefinedtrue%
217   \reledmac@warning{Reference `#1:start' and/or `#1:end' on page \thepage\
space undefined.
218   Using `??'.}%
219   \@latex@warning{Reference `#1:start' and/or `#1:end' undefined\on@line}%
220 }
221 %
222 %

```

```

\led@warn@NoMarginpars223 \newcommand*{\led@warn@NoMarginpars}{%
224   \reledmac@warning{You can't use \string\marginpar\space in numbered text
}}
225   %

```

```

\led@warn@BadSidenotemargin226 \newcommand*{\led@warn@BadSidenotemargin}{%
227   \reledmac@warning{Bad \string\sidenotemmargin\space argument}}
228   %

```

```

\led@warn@NoIndexFile229 \newcommand*{\led@warn@NoIndexFile}[1]{%
230   \reledmac@warning{Undefined index file #1}}
231   %

```

```

\led@warn@SeriesStillExist232 \newcommand{\led@warn@SeriesStillExist}[1]{%
233   \reledmac@warning{Series #1 is still existing !}%
234 }%
235 %

```



```

\led@err@BadAction236 \newcommand*{\led@err@StanzaIndentNotDefined}{%
237 \reledmac@error{You have not defined the indentation for the line \number
\stanza@count}{\@ehc}}%
238 %

\led@err@ManySidenotes239 \newcommand{\led@err@ManySidenotes}{%
\led@err@ManyLeftnotes240 \ifledRcol{%
\led@err@ManyRightnotes241 \reledmac@warning{\itemcount@space sidenotes on line \the\line@numR\
space p. \the\page@numR}%
242 \else%
243 \reledmac@warning{\itemcount@space sidenotes on line \the\line@num\
space p. \the\page@num}%
244 \fi%
245 }%
246 \newcommand{\led@err@ManyLeftnotes}{%
247 \ifledRcol{%
248 \reledmac@warning{\itemcount@space leftnotes on line \the\line@numR\
space p. \the\page@numR}%
249 \else%
250 \reledmac@warning{\itemcount@space leftnotes on line \the\line@num\
space p. \the\page@num}%
251 \fi%
252 }%
253 \newcommand{\led@err@ManyRightnotes}{%
254 \ifledRcol{%
255 \reledmac@warning{\itemcount@space rightnotes on line \the\line@numR\
space p. \the\page@numR}%
256 \else%
257 \reledmac@warning{\itemcount@space rightnotes on line \the\line@num\
space p. \the\page@num}%
258 \fi%
259 }%
260 %

\led@err@TooManyColumns261 \newcommand*{\led@err@TooManyColumns}{%
\led@err@UnequalColumns262 \reledmac@error{Too many columns}{\@ehc}}
\led@err@LowStartColumn263 \newcommand*{\led@err@UnequalColumns}{%
\led@err@HighEndColumn264 \reledmac@error{Number of columns is not equal to the number
\led@err@ReverseColumns265 in the previous row (or \protect\\ \space forgotten?)}{\@ehc}}
266 \newcommand*{\led@err@LowStartColumn}{%
267 \reledmac@error{Start column is too low}{\@ehc}}
268 \newcommand*{\led@err@HighEndColumn}{%
269 \reledmac@error{End column is too high}{\@ehc}}
270 \newcommand*{\led@err@ReverseColumns}{%
271 \reledmac@error{Start column is greater than end column}{\@ehc}}
272 %

```

```

\led@err@toendnotes@outsidenumbering73 \newcommand{\led@err@toendnotes@outsidenumbering}{%
274   \reledmac@error{\string\toendnotes\space and related commands must be
      called inside a numbered text (\string\...beginnumbering\string\endnumbering
      )}{\@ehc}%
275 }%
276 %

```

```

\led@err@EdtextWithoutFootnote77 \newcommand{\led@err@EdtextWithoutFootnote}{%
278   \reledmac@error{edtext without Xfootnote. Check syntax}{\@ehc}%
279 }%
280 %

```

```

\led@err@FootnoteNotInSecondArgEdtext81 \newcommand{\led@err@FootnoteNotInSecondArgEdtext}[1]{%
282   \reledmac@error{#1 footnote outside of the second argument of an edtext.
      Check syntax}{\@ehc}%
283 }%
284 %

```

```

\led@error@PackageAfterEledmac85 \newcommand{\led@error@PackageAfterEledmac}[1]{%
286   \reledmac@error{#1 must be loaded before reledmac}{\@ehc}%
287 }%
288 %

```

```

\led@error@fail@patch@@makecol89 \newcommand{\led@error@fail@patch@@makecol}{%
290   \reledmac@error{Fail to patch \string\@makecol\space command}{\@ehc}%
291 }%
292 %

```

```

\led@error@fail@patch@@reinserts93 \newcommand{\led@error@fail@patch@@reinserts}{%
294   \reledmac@error{Fail to patch \string\@reinserts\space command}{\@ehc}%
295 }%
296 %

```

```

\led@error@fail@patch@@docclearpage97 \newcommand{\led@error@fail@patch@@docclearpage}{%
298   \reledmac@error{Fail to patch \string\@docclearpage\space command}{\@ehc}%
299 }%
300 %

```

```

\led@error@fail@patch@@iiiminipage01 \newcommand{\led@error@fail@patch@@iiiminipage}{%
302   \reledmac@error{Fail to patch \string\@iiiminipage\space command}{\@ehc}%
303 }%
304 %

```

```

or@fail@patch@endminipage 305 \newcommand{\led@error@fail@patch@endminipage}{%
306 \reledmac@error{Failed to patch the \string\endminipage\space command}{\
@ehc}%
307 }%
308 %

```

```

or@fail@patch@endminipage 309 \newcommand{\led@error@fail@patch@makeindex}{%
310 \reledmac@error{Failed to patch the \string\makeindex\space command}{\
@ehc}%
311 }%
312 %

```

```

n@edinde@outsidenumbering 313 \newcommand{\led@warn@edinde@outsidenumbering}{%
314 \reledmac@warning{\string\edindex\space called outside of \string\
...beginnumbering\string\endnumbering. \MessageBreak Automatically switched
to \string\index.}%
315 }%
316 %

```

```

warning@hsizeX@deprecated 317 \newcommand{\led@warning@hsizeX@deprecated}{%
318 \reledmac@warning{\string\hsizeX\space command deprecated, use \string\
widthX\space instead.}%
319 }%
320 %

```

```

warning@Xhsize@deprecated 321 \newcommand{\led@warning@Xhsize@deprecated}{%
322 \reledmac@warning{\string\Xhsize\space command deprecated, use \string\
Xwidth\space instead.}%
323 }%
324 %

```

```

warning@msdatawithoutstop 325 \newcommand{\led@warning@msdatawithoutstop}{%
326 \reledmac@warning{\string\msdata\space without corresponding \string\
stopmsdata}%
327 }%
328 %

```

```

ning@preXnotes@deprecated 329 \newcommand{\led@warning@preXnotes@deprecated}{%
330 \reledmac@warning@preXnotes@deprecated%
331 }%
332 %

```

## II.8 Gobbling

Here, we define some commands which gobble their arguments.

```
\@gobblethree33 \providecommand*\@gobblethree}[3]{}
\@gobblefour34 \providecommand*\@gobblefour}[4]{}
\@gobbleseven35 \providecommand*\@gobbleseven}[7]{}
336 %
```

## II.9 Miscellaneous commands

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

```
337 \ifledfinal
338   \newcommand*\showlemma[1]{#1}
339 \else
340   \newcommand*\showlemma[1]{\underline{#1}}
341 \fi
342
343 %
```

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

```
344 \let\linenumberlist=\empty
345
346 %
```

`\@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).

```
347 \newcount\@l@tempcnta \newcount\@l@tempcntb
348 %
```

## II.10 Prepare reledpar

`\ifl@dpairing` 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@dpaging`  
`\ifl@dprintingpages`

```
\ifl@dprintingcolumns349 \newif\ifl@dpairing
\ifpst@rtedL350 \newif\ifl@dpaging%
\l@dnumpstartsL351 \newif\ifl@dprintingpages%
352 \newif\ifl@dprintingcolumns%
353 \newif\ifpst@rtedL
```

```

354 \newcount\l@dnumpstartsL
355 %

```

`\ifledRcol` `\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`.

```

356 \newif\ifledRcol
357 \newif\ifledRcol@
358 %

```

`\ifnumberingR` The `\ifnumberingR` flag is set to true if we're within a right text numbered section.

```

359 \newif\ifnumberingR
360 %

```

The `\ifXnote@` macro is set to true when we are typesetting a critical footnote.

```

361 \newif\ifXnote@%
362 %

```

## II.11 Booleans provided by other optional packages which are required in any case

`\ifindtl@innote` `\ifindtl@innote` and `\ifindtl@notenumber` are required even if `indextools` is not used.

```

363 \providebool{indtl@innote}%
364 \providebool{indtl@notenumber}%
365 %

```

## 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 `\resumenumbering` 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.

```

366 \newcount\section@num
367 \section@num=0
368 \let\extensionchars=\empty
369 %

```

`\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.

```

370 \newif\ifnumbering
371 %

```

`\beginnumbering` `\initnumbering@reg` `\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@dnumpstartsL` — the number of chunks to be processed.
- set `\ifpst@rtedL` to FALSE.

```

372 \newcommand*{\beginnumbering}{%%
373   \ifnumbering
374     \led@err@NumberingStarted
375   \endnumbering
376 \fi
377 \global\numberingtrue
378 \global\advance\section@num \@ne
379 \initnumbering@reg
380 \message{Section \the\section@num }%
381 \line@list@stuff{\jobname.\extensionchars\the\section@num}%
382 \l@dend@stuff
383 \setcounter{pstart}{1}
384 \ifl@dpairing
385   \global\l@dnumpstartsL \z@
386   \global\pst@rtedLfalse
387 %

```

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.

```

388 \else
389   \beginngroup
390   \global\@afterindenttrue%In order to reestablish normal feature if the \
beginngroup was not here
391   \initnumbering@quote
392   \ifwidthliketwocolumns%
393     \csuse{setwidthliketwocolumns@\columns@position}%
394     \csuse{setpositionliketwocolumns@\columns@position}%
395   \fi%
396 \fi
397 \gdef\eled@sections@{ }%
398 \if@noeled@sec\else%
399   \makeatletter%
400   \InputIfFileExists%
401     {\l@auxdir\jobname.eledsec\the\section@num}%
402     {}%
403     {\led@warn@NoFile{\l@auxdir\jobname.eledsec\the\section@num}}%
404   \makeatother%
405   \immediate\openout\eled@sectioning@out=\l@auxdir\jobname.eledsec\the\
section@num\relax%
406   \fi%
407 }
408 \newcommand*{\initnumbering@reg}{%
409   \global\pst@rtedLfalse
410   \global\l@dnumstartsl \z@
411   \global\absline@num \z@
412   \gdef\normal@page@break{}
413   \gdef\l@prev@pb{}
414   \gdef\l@prev@nopb{}
415   \global\line@num \z@
416   \global\subline@num \z@
417   \global\@lock \z@
418   \global\sub@lock \z@
419   \global\sublines@false
420   \global\let\next@page@num=\relax
421   \global\let\sub@change=\relax
422   \resetprevline@
423   \resetprevpage@num
424   \global\stopmsdata@inserted@true%
425   \global\let\@msdata@list\relax%
426   \global\csundef{\@msdata@\add@msdc @data}%
427 }
428
429 %

```

`\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.

```

430 \def\endnumbering{%
431   \ifnumbering
432     \global\numberingfalse
433     \normal@pars
434     \ifnum\l@dnumstartsL=0%
435       \led@err@NumberingWithoutPstart%
436     \fi%
437     \ifl@dpairing
438       \global\pst@rtedLfalse
439     \else
440       \ifx\insertlines@list\empty\else
441         \global\noteschanged@true
442       \fi
443       \ifx\line@list\empty\else
444         \global\noteschanged@true
445       \fi
446     \fi
447     \ifnoteschanged@
448       \led@mess@NotesChanged
449     \fi
450   \else
451     \led@err@NumberingNotStarted
452   \fi
453   \autoparfalse
454   \if@noeled@sec\else%
455     \immediate\closeout\eled@sectioning@out%
456   \fi%
457   \ifl@dpairing\else
458     \global\l@dnumstartsL=\z@%
459   \endgroup
460 \fi
461 }
462 %

```

`\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>24</sup>

`\resumenumbering`

```

463 \newcommand{\pausenumbering}{%
464   \ifautopar\global\autopar@pausetrue\fi%
465   \endnumbering\global\numberingtrue}
466 %

```

The `\resumenumbering` macro is a bit more involved, but not much. It does most of the same things as `\beginnumbering`, but without resetting the various counters. Note

<sup>24</sup>Peter Wilson's thanks to Wayne Sullivan, who suggested the idea behind these macros.



that no check is made by `\resumenumbering` to ensure that `\pausenumbering` was actually invoked.

```

467 \newcommand*{\resumenumbering}{%
468   \ifnumbering
469     \ifautopar@pause\autopar\fi
470     \global\pstarttrue
471     \global\advance\section@num \@ne
472     \led@mess@SectionContinued{\the\section@num}%
473     \line@list@stuff{\jobname.\extensionchars\the\section@num}%
474     \ledend@stuff
475     \ifl@pairing\else%
476       \begingroup%
477       \initnumbering@quote%
478       \ifwidthliketwocolumns%
479         \csuse{setwidthliketwocolumns@\columns@position}%
480         \csuse{setpositionliketwocolumns@\columns@position}%
481       \fi%
482     \fi%
483     \ifcontinuousnumberingwithcolumns%
484       \ifdefined\line@numR%
485         \ifnum\line@numR>\line@num%
486           \expandafter\setlinenum\expandafter{\the\line@numR}%
487         \fi%
488         \ifnum\last@page@numR>\last@page@num%
489           \global\last@page@num=\last@page@numR%
490         \fi%
491       \fi%
492     \fi%
493   \else
494     \led@err@NumberingShouldHaveStarted
495     \endnumbering
496     \beginnumbering
497   \fi}
498
499 %
500 %

```

## 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  $\LaTeX$ 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.

```
501 \newcommand*\list@create}[1]{%
502   \global\let#1=\empty%
503 }%
504 %
```

**\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 .

```
505 \newcommand*\list@clear}[1]{%
506   \global\let#1=\empty%
507 }
508 %
```

**\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`.

```
509 \newtoks\led@toksa \newtoks\led@toksb
510 \global\led@toksa={\}
511 \long\def\xright@appenditem#1\to#2{%
512   \global\led@toksb=\expandafter{#2}%
513   \xdef#2{\the\led@toksb\the\led@toksa\expandafter{#1}}%
514   \global\led@toksb={}}
515 %
```

**\xleft@appenditem** `\xleft@appenditem` expands an item and appends it to the left end of a list macro; it is otherwise identical to `\xright@appenditem`.

```
516 \long\def\xleft@appenditem#1\to#2{%
517   \global\led@toksb=\expandafter{#2}%
518   \xdef#2{\the\led@toksa\expandafter{#1}\the\led@toksb}%
519   \global\led@toksb={}}
520 %
```

**\gl@p** The `\gl@p` macro removes the leftmost item from a list and places it in a control sequence. You type `\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.

```

521 \def\gl@p#1\to#2{\expandafter\gl@poff#1\gl@poff#1#2}
522 \long\def\gl@poff\#1#2\gl@poff#3#4{\gdef#4{#1}\gdef#3{#2}}
523
524 %

```

## 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 specify the current lineation system:

- line-of-page: `bypstart@ = false` and `bypage@ = true`.
- line-of-pstart: `bypstart@ = true` and `bypage@ = false`.

`\ifbypage@`  
`\bypage@true`  
`\bypage@false` reledmac will use the line-of-section system unless instructed otherwise.

```

525 \newif\ifbypage@
526 \newif\ifbypstart@
527 %

```

The `\ifbypage@R` and `\ifbypstart@R` flag specify 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@R28 \newif\ifbypage@R
\ifbypstart@R29 \newif\ifbypstart@R
530 %

```

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

```

531 \newcommand*{\lineation}[1]{
532 %

```

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

```

533 \ifnumbering
534 \led@err@LineationInNumbered
535 \else
536 %

```

If the argument is page.

```

537 \def\@tempa{#1}\def\@tempb{page}%
538 \ifx\@tempa\@tempb
539   \global\bypage@true
540   \global\bystart@false
541   \unless\ifnocritical@%
542     \Xpstart[] [false]%
543   \fi%
544 %

```

If the argument is pstart.

```

545 \else
546   \def\@tempb{pstart}%
547   \ifx\@tempa\@tempb
548     \global\bypage@false
549     \global\bystart@true
550     \unless\ifnocritical@%
551       \Xpstart%
552     \fi%
553 %

```

And finally, if the argument is section (default).

```

554 \else
555   \def\@tempb{section}%
556   \ifx\@tempa\@tempb
557     \global\bypage@false
558     \global\bystart@false
559     \unless\ifnocritical@%
560       \Xpstart[] [false]%
561     \fi%
562 %

```

In other case, it is an error.

```

563 \else
564   \led@warn@BadLineation
565 \fi
566 \fi
567 \fi
568 \fi}}
569 %

```

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

```

570 \newcount\line@margin%
571 \newcount\line@margin@columns%Only for parallel typesetting

```

```

572 \line@margin@columns=\m@ne%
573
574 \newcommand*{\linenummargin}[1]{%
575   \l@getline@margin{#1}%
576   \ifnum\@l@dttempcntb>\m@ne
577     \ifledRcol
578       \global\line@marginR=\@l@dttempcntb
579       \led@warn@setting@in@rightside{\linenummargin}%
580     \else
581       \global\line@margin=\@l@dttempcntb
582     \fi
583   \fi}}
584
585 \newcommand*{\l@getline@margin}[1]{%
586   \def\@tempa{#1}\def\@tempb{left}%
587   \ifx\@tempa\@tempb
588     \@l@dttempcntb \z@
589   \else
590     \def\@tempb{right}%
591     \ifx\@tempa\@tempb
592       \@l@dttempcntb \@ne
593     \else
594       \def\@tempb{outer}%
595       \ifx\@tempa\@tempb
596         \@l@dttempcntb \tw@
597       \else
598         \def\@tempb{inner}%
599         \ifx\@tempa\@tempb
600           \@l@dttempcntb \thr@@
601         \else
602           \led@warn@BadLinenummargin
603           \@l@dttempcntb \m@ne
604         \fi
605       \fi
606     \fi
607   \fi}}
608
609 %

```

### 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`    `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.

```

610 \newcounter{firstlinenum}
611 \setcounter{firstlinenum}{5}

```

```

612 \newcounter{linenumincrement}
613 \setcounter{linenumincrement}{5}
614 %

```

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

```

615 \newcounter{firstsublinenum}
616 \setcounter{firstsublinenum}{5}
617 \newcounter{sublinenumincrement}
618 \setcounter{sublinenumincrement}{5}
619
620 %

```

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

```

\linenumincrement
\firstsublinenum
\sublinenumincrement
621 \newcommand*{\firstlinenum}[1]{%
622   \ifledRcol%
623     \setcounter{firstlinenumR}{#1}%
624     \led@warn@setting@in@rightside{\firstlinenum}%
625   \else%
626     \setcounter{firstlinenum}{#1}%
627   \fi%
628 }
629 \newcommand*{\linenumincrement}[1]{%
630   \ifledRcol%
631     \setcounter{linenumincrementR}{#1}%
632     \led@warn@setting@in@rightside{\linenumincrement}%
633   \else%
634     \setcounter{linenumincrement}{#1}%
635   \fi%
636 }
637 \newcommand*{\firstsublinenum}[1]{%
638   \ifledRcol%
639     \setcounter{firstsublinenumR}{#1}%
640     \led@warn@setting@in@rightside{\firstsublinenum}%
641   \else%
642     \setcounter{firstsublinenum}{#1}%
643   \fi%
644 }
645 \newcommand*{\sublinenumincrement}[1]{%
646   \ifledRcol%
647     \setcounter{sublinenumincrementR}{#1}%
648     \led@warn@setting@in@rightside{\sublinenumincrement}%
649   \else%
650     \setcounter{sublinenumincrement}{#1}%
651   \fi%
652 }
653
654

```

655 %

## V.4 Line number locking

`\lockdisp` When line locking is being used, the `\lockdisp{⟨word⟩}` macro specifies whether a line number—if one is due to appear—should be printed on the first printed line or on the last, or by all of them. Its argument is a word, either `first`, `last`, or `all`. Initially, it is set to `first`.

`\lock@disp` encodes the selection: 0 for first, 1 for last, 2 for all.

```

656 \newcount\lock@disp
657 \newcommand{\lockdisp}[1]{%
658   \l@dgetlock@disp{#1}%
659   \ifnum\l@dtempcntb>\m@ne
660     \global\lock@disp=\l@dtempcntb
661   \else
662     \led@warn@BadLockdisp
663   \fi}}
664 \newcommand*{\l@dgetlock@disp}[1]{
665   \def\@tempa{#1}\def\@tempb{first}%
666   \ifx\@tempa\@tempb
667     \l@dtempcntb \z@
668   \else
669     \def\@tempb{last}%
670     \ifx\@tempa\@tempb
671       \l@dtempcntb \@ne
672     \else
673       \def\@tempb{all}%
674       \ifx\@tempa\@tempb
675         \l@dtempcntb \tw@
676       \else
677         \l@dtempcntb \m@ne
678       \fi
679     \fi
680   \fi}
681
682 %

```

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

```

683 \newcount\sublock@disp
684 \newcommand{\sublockdisp}[1]{%
685   \l@dgetlock@disp{#1}%
686   \ifnum\l@dtempcntb>\m@ne
687     \global\sublock@disp=\l@dtempcntb
688   \else
689     \led@warn@BadSublockdisp
690   \fi}}

```

```
691
692 %
```

## V.5 Line number style

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

```
693 \newcommand*{\linenumberstyle}[1]{%
694   \def\linenumrep##1{\@nameuse{##1}}
695 \newcommand*{\sublinenumberstyle}[1]{%
696   \def\sublinenumrep##1{\@nameuse{##1}}
697 %
```

Initialise the number styles to arabic.

```
698 \linenumberstyle{arabic}
699 \let\linenumr@p\linenumrep
700 \sublinenumberstyle{arabic}
701 \let\sublinenumr@p\sublinenumrep
702
703 %
```

## V.6 Line number printing

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

The original `\numlabfont` specification is equivalent to the  $\text{\LaTeX}$  `\scriptsize` for a 10pt document.

```
704 \newlength{\linenumsep}
705 \setlength{\linenumsep}{1pc}
706 \newcommand*{\numlabfont}{\normalfont\scriptsize}
```



```

707 \newcommand*{\ledlinenum}{%
708   \bgroup%
709   \ifluatex%
710     \texdir TLT%
711   \fi%
712   \numlabfont\linenumrep{\line@num}%
713   \ifsublines@
714     \ifnum\subline@num>0\relax
715       \unskip%
716       \Xsublinesep@side%
717       \sublinenumrep{\subline@num}%
718     \fi
719   \fi%
720   \egroup%
721 }%
722
723 \newcommand*{\leftlinenum}{%
724   \ledlinenum
725   \kern\linenumsep}
726 \newcommand*{\rightlinenum}{%
727   \kern\linenumsep
728   \ledlinenum}
729
730 %

```

## 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.

**`\line@num`** The count `\line@num` 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 `\subline@num`.

```

731 \newcount\line@num
732 %

```

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

```
733 \newcount\subline@num
734 %
```

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

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

`\sublines@false`

```
735 \newif\ifsublines@
736 %
```

`\absline@num` The count `\absline@num` 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 `\line@num` will often be equal to it. It is used internally to keep track of where notes are to appear and where new pages start: using this value rather than `\line@num` is a lot simpler, because it does not depend on the lineation system in use.

```
737 \newcount\absline@num
738 %
```

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.

`\sub@lock`

```
739 \newcount\@lock
740 \newcount\sub@lock
741 %
```

## 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|0T1/cmr/m/n.
```

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

- `\insertlines@list`: the line numbers of lines that have footnotes or other insertions. These are the absolute numbers where the corresponding lemmas begin. This list contains one entry for every footnote in the section; one lemma may contribute no footnotes or many footnotes. This list is used by `\add@inserts` within `\do@line`, to tell it where to insert notes.
- `\actionlines@list`: a list of absolute line numbers at which we are to perform special actions; these actions are specified by the `\actions@list` list defined below.
- `\actions@list`: action codes corresponding to the line numbers in `\actionlines@list`. These codes tell `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:

```

742 \list@create{\line@list}
743 \list@create{\insertlines@list}
744 \list@create{\actionlines@list}
745 \list@create{\actions@list}
746
747 %
```

`\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`

```

748 \newcount\page@num
749 \newcount\endpage@num
750 \newcount\endline@num
751 \newcount\endsubline@num
752 %

```

`\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.

```

753 \newif\ifnoteschanged@
754 %

```

`\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@55 \newcommand*\resetprevline@{%
756 \def\do##1{\global\csundef{prevline##1}}%
757 \dolistloop{\@series}%
758 }
759 %

```

`\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@50 \newcommand*\resetprevpage@num{%
761 \def\do##1{%
762 \ifcsdef{prevpage##1@num}{%
763 \global\csname prevpage##1@num\endcsname=\z@%
764 \global\csname prevpage##1@numR\endcsname=\z@%
765 }%
766 {}%
767 \ifcsdef{##1prevpage@num}{%

```

```

768     \global\csname ##1prevpage@num\endcsname=\z@%
769     \global\csname ##1prevpage@numR\endcsname=\z@%
770     }%
771     {}%
772     }%
773     \dolistloop{\@series}%
774 }
775 %

```

## 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.

```

776 \newread\@inputcheck
777 \newcommand*{\read@linelist}[1]{%
778     \ifledRcol%
779     \list@clearing@regR%
780     \else%
781     \list@clearing@reg%
782     \fi%
783 %

```

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).

```

784     \list@clear{\maxlinesinpar@list}
785 %

```

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.

```

786 \get@linelistfile{#1}%
787 \ifcontinuousnumberingwithcolumns
788     \global\page@numR=\page@numR\relax
789     \global\last@page@numR=\last@page@numR\relax

```

```

790 \global\page@num=\page@num\relax
791 \global\last@page@num=\last@page@num\relax
792 \fi
793 \@stopmsd%Security if last \endms{} is forgotten
794 \unless\ifledRcol%Get the last line of the last page
795 \cnumgdef{@lastabsline@forpage@\the\page@num}{\the\absline@num}%
796 \cnumgdef{@lastline@forpage@\the\page@num}{\the\line@num}%
797 \else%
798 \cnumgdef{@lastabsline@forpageR@\the\page@numR}{\the\absline@numR}%
799 \cnumgdef{@lastline@forpageR@\the\page@numR}{\the\line@numR}%
800 \fi%
801 \endgroup
802 %

```

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.

```

803 \ifledRcol
804 \global\page@numR=\m@ne
805 \ifx\actionlines@listR\empty
806 \gdef\next@actionlineR{1000000}%
807 \else
808 \gl@p\actionlines@listR\to\next@actionlineR
809 \gl@p\actions@listR\to\next@actionR
810 \fi
811 \else
812 \global\page@num=\m@ne
813 \ifx\actionlines@list\empty
814 \gdef\next@actionline{1000000}%
815 \else
816 \gl@p\actionlines@list\to\next@actionline
817 \gl@p\actions@list\to\next@action
818 \fi
819 \fi
820 }
821 %

```

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

```

822 \newcommand*{\list@clearing@reg}{%
823 \list@clear{\line@list}%
824 \list@clear{\insertlines@list}%
825 \list@clear{\actionlines@list}%
826 \list@clear{\actions@list}%
827 \list@clear{\linesinpar@listL}%
828 \list@clear{\linesonpage@listL}%
829 }%
830 %

```

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

```

831 \newcommand*\get@linelistfile}[1]{%
832   \InputIfFileExists{\l@auxdir#1}{%
833     \global\noteschanged@false
834     \begingroup
835       \catcode`\[=1 \catcode`\]=2
836       \makeatletter \catcode`\^^M=9}{%
837     \led@warn@NoFile{\l@auxdir#1}%
838     \global\noteschanged@true
839     \begingroup}%
840 }
841
842 %

```

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 5.2.7 p. 20 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 use `\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.

```

843 \newcommand{\line@list@version}[1]{%
844   \IfStrEq{#1}{\this@line@list@version}%
845   {}%

```



```

846 {\ifledRcol%
847   \led@warn@Obsolete{\jobname.\extensionchars\the\section@num}%
848   \else%
849   \led@warn@Obsolete{\jobname.\extensionchars\the\section@num}%
850   \fi%
851   \endinput%
852 }%
853 }%
854 %

```

**\@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.

```

855
856 \newcommand*{\@nl}[2]{%
857   \fix@page{#1}%
858   \ifledRcol%
859   \@nl@regR%
860   \else%
861   \@nl@reg%
862   \fi%
863 }
864 \newcommand*{\@nl@reg}{%
865   \ifx\l@dchset@num\relax \else
866   \advance\absline@num \@ne
867   \set@line@action
868   \let\l@dchset@num=\relax
869   \advance\absline@num \m@ne
870   \advance\line@num \m@ne
871   \fi
872 %

```

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

```

873 \advance\absline@num \@ne
874 \ifx\next@page@num\relax \else
875   \page@action

```

```

876         \let\next@page@num=\relax
877     \fi
878     \ifx\sub@change\relax \else
879         \ifnum\sub@change>\z@
880             \sublines@true
881         \else
882             \sublines@false
883         \fi
884     \sub@action
885     \let\sub@change=\relax
886 \fi
887 %

```

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

```

888     \ifcase\@lock
889     \or
890         \@lock \tw@
891     \or \or
892         \@lock \z@
893     \fi
894     \ifcase\sub@lock
895     \or
896         \sub@lock \tw@
897     \or \or
898         \sub@lock \z@
899     \fi
900 %

```

Now advance the visible line number, unless it has been locked.

```

901     \ifsublines@
902         \ifnum\sub@lock<\tw@
903             \advance\subline@num \@ne
904         \fi
905     \else
906         \ifnum\@lock<\tw@
907             \advance\line@num \@ne \subline@num \z@
908         \fi
909     \fi}
910
911 %

```

`\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`.

```

912 \newcount\last@page@num
913 \last@page@num=-10000
914
915 \newcommand*{\fix@page}[1]{%
916     \ifledRcol

```

```

917 \ifnum #1=\last@page@numR
918 \else
919 \csnumgdef{@lastabsline@forpageR@the\page@numR}{\the\absline@numR}%
920 \csnumgdef{@lastline@forpageR@the\page@numR}{\the\line@numR}%
921 \ifbypage@R
922 \line@numR \z@ \subline@numR \z@
923 \fi
924 \page@numR=#1\relax
925 \last@page@numR=#1\relax
926 \def\next@page@numR{#1}%
927 \fi
928 \else
929 \ifnum #1=\last@page@num
930 \else
931 \csnumgdef{@lastabsline@forpage@the\page@num}{\the\absline@num}%
932 \csnumgdef{@lastline@forpage@the\page@num}{\the\line@num}%
933 \ifbypage@
934 \line@num \z@ \subline@num \z@
935 \fi
936 \page@num=#1\relax
937 \last@page@num=#1\relax
938 \def\next@page@num{#1}%
939 \listxadd{\normal@page@break}{\the\absline@num}
940 \fi
941 \fi}
942 %

```

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

**\@lopL**  
**\@lopR**

```

943 \newcommand*{\@pend}[1]{}
944 \newcommand*{\@pendR}[1]{}
945 \newcommand*{\@lopL}[1]{}
946 \newcommand*{\@lopR}[1]{}
947
948 %

```

**\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 **\@n1** of the necessary action.

```

949 \newcommand*{\sub@on}{\ifsublines@
950 \let\sub@change=\relax
951 \else
952 \def\sub@change{1}%
953 \fi}
954 \newcommand*{\sub@off}{\ifsublines@
955 \def\sub@change{-1}%
956 \else

```

```

957 \let\sub@change=\relax
958 \fi}
959
960 %

```

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

```

961
962 \newcommand*{\@adv}[1]{%
963   \ifsublines@
964     \ifledRcol
965       \advance\subline@numR by #1\relax
966       \ifnum\subline@numR<\z@
967         \led@warn@BadAdvancelineSubline
968         \subline@numR \z@
969       \fi
970     \else
971       \advance\subline@num by #1\relax
972       \ifnum\subline@num<\z@
973         \led@warn@BadAdvancelineSubline
974         \subline@num \z@
975       \fi
976     \fi
977   \else
978     \ifledRcol
979       \advance\line@numR by #1\relax
980       \ifnum\line@numR<\z@
981         \led@warn@BadAdvancelineLine
982         \line@numR \z@
983       \fi
984     \else
985       \advance\line@num by #1\relax
986       \ifnum\line@num<\z@
987         \led@warn@BadAdvancelineLine
988         \line@num \z@
989       \fi
990     \fi
991   \fi
992   \set@line@action}
993
994 %

```

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

```

995
996 \newcommand*{\@set}[1]{%
997   \ifledRcol
998     \ifsublines@

```

```

999      \subline@numR=#1\relax
1000    \else
1001      \line@numR=#1\relax
1002    \fi
1003    \set@line@action
1004  \else
1005    \ifsublines@
1006      \subline@num=#1\relax
1007    \else
1008      \line@num=#1\relax
1009    \fi
1010    \set@line@action
1011  \fi}
1012
1013 %

```

**\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.

```

1014
1015 \newcommand*{\l@d@set}[1]{%
1016   \ifledRcol
1017     \line@numR=#1\relax
1018     \advance\line@numR \@ne
1019     \def\l@dchset@num{#1}
1020   \else
1021     \line@num=#1\relax
1022     \advance\line@num \@ne
1023     \def\l@dchset@num{#1}
1024   \fi}
1025 \let\l@dchset@num\relax
1026
1027 %

```

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

```

1028
1029 \newcommand*{\page@action}{%
1030   \ifledRcol
1031     \xright@appenditem{\the\absline@numR}\to\actionlines@listR
1032     \xright@appenditem{\next@page@numR}\to\actions@listR
1033   \else
1034     \xright@appenditem{\the\absline@num}\to\actionlines@list
1035     \xright@appenditem{\next@page@num}\to\actions@list
1036   \fi}
1037 %

```

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

```

1038
1039 \newcommand*{\set@line@action}{%
1040   \ifledRcol
1041     \xright@appenditem{\the\absline@numR}\to\actionlines@listR
1042     \ifsublines@
1043       \@l@tempcnta=-\subline@numR
1044     \else
1045       \@l@tempcnta=-\line@numR
1046     \fi
1047     \advance\@l@tempcnta by -5000\relax
1048     \xright@appenditem{\the\@l@tempcnta}\to\actions@listR
1049   \else
1050     \xright@appenditem{\the\absline@num}\to\actionlines@list
1051     \ifsublines@
1052       \@l@tempcnta=-\subline@num
1053     \else
1054       \@l@tempcnta=-\line@num
1055     \fi
1056     \advance\@l@tempcnta by -5000\relax
1057     \xright@appenditem{\the\@l@tempcnta}\to\actions@list
1058   \fi}
1059 %

```

**\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.

```

1060
1061 \newcommand*{\sub@action}{%
1062   \ifledRcol
1063     \xright@appenditem{\the\absline@numR}\to\actionlines@listR
1064     \ifsublines@
1065       \xright@appenditem{-1001}\to\actions@listR
1066     \else
1067       \xright@appenditem{-1002}\to\actions@listR
1068     \fi
1069   \else
1070     \xright@appenditem{\the\absline@num}\to\actionlines@list
1071     \ifsublines@
1072       \xright@appenditem{-1001}\to\actions@list
1073     \else
1074       \xright@appenditem{-1002}\to\actions@list
1075     \fi
1076   \fi}
1077 %

```

**\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.

**\do@lockon**

**\do@lockonL**

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.

```

1078 \newcommand*{\lock@on}{\futurelet\next\do@lockon}
1079
1080 \newcommand*{\do@lockon}{%
1081   \ifx\next\lock@off
1082     \global\let\lock@off=\skip@lockoff
1083   \else
1084     \ifledRcol
1085       \do@lockonR
1086     \else
1087       \do@lockonL
1088     \fi
1089   \fi}
1090
1091
1092 \newcommand*{\do@lockonL}{%
1093   \xright@appenditem{\the\absline@num}\to\actionlines@list
1094   \ifsublines@
1095     \xright@appenditem{-1005}\to\actions@list
1096     \ifnum\sub@lock=\z@
1097       \sub@lock \@ne
1098     \else
1099       \ifnum\sub@lock=\thr@@
1100         \sub@lock \@ne
1101       \fi
1102     \fi
1103   \else
1104     \xright@appenditem{-1003}\to\actions@list
1105     \ifnum\@lock=\z@
1106       \@lock \@ne
1107     \else
1108       \ifnum\@lock=\thr@@
1109         \@lock \@ne
1110       \fi
1111     \fi
1112   \fi}
1113
1114 %

```

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

```

\do@lockoff
\do@lockoffL
\skip@lockoff
1115 \newcommand*{\do@lockoffL}{%
1116   \xright@appenditem{\the\absline@num}\to\actionlines@list
1117   \ifsublines@
1118     \xright@appenditem{-1006}\to\actions@list
1119     \ifnum\sub@lock=\tw@

```

```

1120     \sub@lock \thr@@
1121     \else
1122     \sub@lock \z@
1123     \fi
1124     \else
1125     \xright@appenditem{-1004}\to\actions@list
1126     \ifnum \@lock=\tw@
1127     \@lock \thr@@
1128     \else
1129     \@lock \z@
1130     \fi
1131     \fi}
1132
1133 \newcommand*{\do@lockoff}{%
1134     \ifledRcol
1135     \do@lockoffR
1136     \else
1137     \do@lockoffL
1138     \fi}
1139 \newcommand*{\skip@lockoff}{\global\let\lock@off=\do@lockoff}
1140 \global\let\lock@off=\do@lockoff
1141
1142 %

```

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

```

1143 \newcommand*{\n@num}{%
1144     \ifledRcol%
1145     \xright@appenditem{\the\absline@numR}\to\actionlines@listR
1146     \xright@appenditem{-1007}\to\actions@listR
1147     \else%
1148     \xright@appenditem{\the\absline@num}\to\actionlines@list%
1149     \xright@appenditem{-1007}\to\actions@list%
1150     \fi%
1151 }%
1152
1153 %

```

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

```

1154 \newcommand*{\n@num@stanza}{%
1155     \ifledRcol%
1156     \xright@appenditem{\the\absline@numR}\to\actionlines@listR%
1157     \xright@appenditem{-1008}\to\actions@listR%
1158     \else%
1159     \xright@appenditem{\the\absline@num}\to\actionlines@list%%
1160     \xright@appenditem{-1008}\to\actions@list%
1161     \fi%

```



```
1162 }
1163 %
```

`\ifl@dhidenumber` `\hidenum`bering hides number in margin. It uses action code 1009. `\hidenum`beringonleftpage and `\hidenum`beringonrightpage are variant, using action code only conditionnaly

```
\hidenumberingonleftpage
\hidenumberingonrightpage
\newif\ifl@dhidenumber
1165 \newcommand*{\hidenumbering}{
1166   \ifl@dhidenumber
1167     \write\linenum@outR{\string\hide@num}%
1168   \else%
1169     \write\linenum@out{\string\hide@num}%
1170   \fi%
1171 }%
1172 \newcommand*{\hide@num}{%
1173   \ifl@dhidenumber
1174     \xright@appenditem{\the\absline@numR}\to\actionlines@listR%
1175     \xright@appenditem{-1009}\to\actions@listR%
1176   \else%
1177     \xright@appenditem{\the\absline@num}\to\actionlines@listR%
1178     \xright@appenditem{-1009}\to\actions@listR%
1179   \fi%
1180 }
1181 \newcommand*{\hidenumberingonleftpage}{%
1182   \ifl@dhidenumber
1183     \write\linenum@outR{\string\hide@num@left}%
1184   \else%
1185     \write\linenum@out{\string\hide@num@left}%
1186   \fi%
1187 }%
1188
1189 \newcommand*{\hide@num@left}{%
1190   \ifl@dhidenumber
1191     \ifodd\page@numR\else%
1192       \xright@appenditem{\the\absline@numR}\to\actionlines@listR%
1193       \xright@appenditem{-1009}\to\actions@listR%
1194     \fi%
1195   \else%
1196     \ifodd\page@num\else%
1197       \xright@appenditem{\the\absline@num}\to\actionlines@listR%
1198       \xright@appenditem{-1009}\to\actions@listR%
1199     \fi%
1200   \fi%
1201 }%
1202
1203 \newcommand*{\hidenumberingonrightpage}{%
1204   \ifl@dhidenumber
1205     \write\linenum@outR{\string\hide@num@right}%
1206   \else%
```

```

1207 \write\linenum@out{\string\hide@num@right}%
1208 \fi%
1209 }%
1210
1211 \newcommand*\hide@num@right}{%
1212 \ifledRcol%
1213 \ifodd\page@numR%
1214 \xright@appenditem{\the\absline@numR}\to\actionlines@listR%
1215 \xright@appenditem{-1009}\to\actions@listR%
1216 \fi%
1217 \else%
1218 \ifodd\page@num%
1219 \xright@appenditem{\the\absline@num}\to\actionlines@list%%
1220 \xright@appenditem{-1009}\to\actions@list%
1221 \fi%
1222 \fi%
1223 }%
1224
1225 %

```

**\@ref** **\insert@count** \@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.

```

1226 \newcount\insert@count
1227 %

```

- #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.

```

1228 \newcommand*\dummy@ref}[2]{#2}
1229 %

```

**\@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.

```

1230 \newcommand*\@ref}[2]{%
1231 \ifledRcol%
1232 \@ref@regR{#1}{#2}%
1233 \else%
1234 \@ref@reg{#1}{#2}%
1235 \fi%
1236 }%

```

```

1237 \newcommand*{\@ref@reg}[2]{%
1238   \global\insert@count=#1\relax
1239   \global\advance\@edtext@level by 1%
1240   \loop\ifnum\insert@count>\z@
1241     \xright@appenditem{\the\absline@num}\to\insertlines@list
1242     \global\advance\insert@count \m@ne
1243   \repeat
1244   %

```

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.

```

1245 \begingroup
1246   \let\@ref=\dummy@ref
1247   \let\@lopL\@gobble
1248   \let\page@action=\relax
1249   \let\sub@action=\relax
1250   \let\set@line@action=\relax
1251   \let\@lab=\relax
1252   \let\@lemma=\relax%
1253   \let\@sw\@gobblethree%
1254   #2
1255   \global\endpage@num=\page@num
1256   \global\endline@num=\line@num
1257   \global\endsubline@num=\subline@num
1258 \endgroup
1259 %

```

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

```

1260   \xright@appenditem%
1261     {\the\page@num|\the\line@num}%
1262     \ifsublines@ \the\subline@num \else 0\fi}%
1263     \the\endpage@num|\the\endline@num}%
1264     \ifsublines@ \the\endsubline@num \else 0\fi}\to\line@list
1265   %

```

And now, call \@ref@reg@parsearg, which can be also called by \@ref@later

```

1266   \@ref@reg@parse{#2}%
1267   %

```

Decrease edtext level counter.

```

1268   \global\advance\@edtext@level by -1%
1269 }
1270 %

```

**\@ref@reg@parse** The \@ref@reg@parsearg command parses the second argument of a \@ref or the unique argument of \@ref@later written in the auxiliary fill.

First, 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.

```

1271 \newcommand{\@ref@reg@parse}[1]{%
1272   \expandafter\list@create\expandafter{\csname sw@list@edtext@tmp@\the\
@edtext@level\endcsname}%
1273   \providebool{lemmacommand@\the\@edtext@level}%
1274   \boolfalse{lemmacommand@\the\@edtext@level}%
1275   %

```

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

```

1276   #1%
1277   %

```

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.

```

1278   \ifnum\@edtext@level>0%
1279     \def\create@this@edtext@level{\expandafter\list@create\expandafter{\
csname sw@list@edtext@\the\@edtext@level\endcsname}}%
1280     \ifcsundef{sw@list@edtext@\the\@edtext@level}{\create@this@edtext@level
}\}%
1281     \letcs{\@tmp}{sw@list@edtext@\the\@edtext@level}%
1282     \letcs{\@tmpp}{sw@list@edtext@tmp@\the\@edtext@level}
1283     \xright@appenditem{\expandonce\@tmpp}\to\@tmp%
1284     \global\cslet{sw@list@edtext@\the\@edtext@level}{\@tmp}%
1285     \fi%
1286   %
1287 }
1288
1289 %

```

**\ref@reg@later** This macro is stored in the auxiliary file when using \edtextlater. It is used only to get the correct value for the \sameword tools.

```

1290 \newcommand{\@ref@later}[1]{%
1291   \global\advance\@edtext@level by \@ne%
1292   \ifledRcol%
1293     \@ref@reg@parseR{#1}%
1294   \else%
1295     \@ref@reg@parse{#1}%
1296   \fi%
1297   \global\advance\@edtext@level by -\@ne%
1298 }%
1299 %

```

## 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`.

```
1300 \newwrite\linenum@out
1301 %
```

`\iffirst@linenum@out@`  
`\first@linenum@out@true`  
`\first@linenum@out@false`

Once any file is opened on this stream, we keep it open forever, or else switch to another file that we keep open. The reason is that we want the output routine to write the page number for every page to this file; otherwise we 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.

```
1302 \newif\iffirst@linenum@out@
1303 \first@linenum@out@true
1304 %
```

`\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.

```
1305 \newcommand{\this@line@list@version}{6}%
1306 %
```

`\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.

```
1307 \newcommand*{\line@list@stuff}[1]{%
1308 %
```

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

```
1309 \read@linelist{#1}%
1310 %
```

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.

```

1311 \iffirst@linenum@out@
1312 \immediate\closeout\linenum@out%
1313 \global\first@linenum@out@false%
1314 \immediate\openout\linenum@out=\l@auxdir#1\relax%
1315 \immediate\write\linenum@out{\string\line@list@version{\
this@line@list@version}}}%
1316 \ifl@dpaging%
1317 \immediate\write\linenum@out{\string\@par@sync@option{\
@par@this@sync@option}}}%
1318 \fi%
1319 \else
1320 %

```

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.

```

1321 \if@minipage%
1322 \leavevmode%
1323 \fi%
1324 \closeout\linenum@out%
1325 \openout\linenum@out=\l@auxdir#1\relax%
1326 \write\linenum@out{\string\line@list@version{\this@line@list@version}}%
1327 %
1328 \ifl@dpaging%
1329 \write\linenum@out{\string\@par@sync@option{\@par@this@sync@option}}%
1330 %
1331 \fi%
1332 \fi}
1333 %

```

**\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.

```

1333 \newcommand*{\new@line}{%
1334 \IfStrEq{\led@pb@setting}{after}%
1335 {\xifinlist{\the\absline@num}{\l@prev@nopb}%
1336 {\xifinlist{\the\absline@num}{\normal@page@break}%
1337 {\numdef{\@next@page}{\c@page+\@ne}%
1338 \write\linenum@out{\string\@nl[\@next@page][\@next@page}}}%
1339 }%
1340 {\write\linenum@out{\string\@nl[\the\c@page][\thepage}}}%
1341 }%
1342 {\write\linenum@out{\string\@nl[\the\c@page][\thepage}}}%
1343 }%
1344 \IfStrEq{\led@pb@setting}{before}%
1345 {\numdef{\next@absline}{\the\absline@num+\@ne}%

```

```

1346 \xifinlist{\next@absline}{\l@prev@nopb}%
1347 {\xifinlist{\the\absline@num}{\normal@page@break}%
1348 {\numgdef{\nc@page}{\c@page+\@ne}%
1349 \write\linenum@out{\string\@nl[\nc@page][\nc@page]}%
1350 }%
1351 {\write\linenum@out{\string\@nl[\the\c@page][\thepage]}}%
1352 }%
1353 {\write\linenum@out{\string\@nl[\the\c@page][\thepage]}}%
1354 }%
1355 {}%
1356 \IfStrEqCase{\led@pb@setting}{\before}{\relax}{\after}{\relax}}{\write\linenum@out{\string\@nl[\the\c@page][\thepage]}}%
1357 }
1358
1359 %

```

**\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 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.

**\flag@end**

```

1360 \newif\if@noneed@Footnote%
1361
1362 \newcommand*{\flag@start}{%
1363   \ifledRcol%
1364     \edef\next{\write\linenum@outR{%
1365       \string\@ref[\the\insert@countR] []}%
1366     \next%
1367     \ifnum\insert@countR<1%
1368       \if@noneed@Footnote\else%
1369         \led@err@EdtextWithoutFootnote%
1370       \fi%
1371     \fi%
1372   \else%
1373     \edef\next{\write\linenum@out{%
1374       \string\@ref[\the\insert@count] []}%
1375     \next%
1376     \ifnum\insert@count<1%
1377       \if@noneed@Footnote\else%
1378         \led@err@EdtextWithoutFootnote%
1379       \fi%
1380     \fi%
1381   \fi}%
1382
1383 \newcommand*{\flag@end}{%
1384   \ifledRcol%
1385     \write\linenum@outR{[]}%
1386   \else%

```

```

1387 \write\linenum@out{}}%
1388 \fi}%
1389
1390
1391 %

```

`\flag@start@RTL` With Xe<sub>La</sub>TeX, there is a problem when using RTL: the writing of a command in the numbered auxiliary files (.1, .2 etc) is reversed when the first argument of `\edtext` is typeset in one line, but it is **not** reversed when this first argument is typeset in two lines or more.<sup>25</sup>

To solve this problem, we use a crossref mechanism. At the first run, we put a label, but we do not write any `\@ref` command. When the value of the label can be tested, that is after three runs, we're doing:

- If the first argument of `\edtext` is typeset on only one line, we first call `\flag@end`, at the point we normally call `\flag@start`, at the beginning of the content of the first argument, and we call `\flag@end` at the point we normally call `\flag@start`, at the end of the content of the first argument.
- If the first argument of `\edtext` is typeset on only two lines, we use the normal order.

This system is a workaround for the problem of order when writing in auxiliary files.

The `\flag@start@RTL` and `\flag@end@RTL` macro put the label, do the test and call the right commands.

```

1392 \newcommand{\flag@start@RTL}{%
1393   \edlabel{edtext:start:\csuse{thisedtext@the\@edtext@level}}%
1394   \IfStrEq{\xabslineref{edtext:start:\csuse{thisedtext@the\@edtext@level}}}{%
1395     {000}%
1396     {}%
1397     {%
1398       \ifnumequal%
1399         {\xabslineref{edtext:start:\csuse{thisedtext@the\@edtext@level}}}{%
1400         {\xabslineref{edtext:end:\csuse{thisedtext@the\@edtext@level}}}%
1401         {\flag@end}%
1402         {\flag@start}%
1403       }%
1404     }%
1405
1406 \newcommand{\flag@end@RTL}{%
1407   \edlabel{edtext:end:\csuse{thisedtext@the\@edtext@level}}%
1408   \IfStrEq{\xabslineref{edtext:start:\csuse{thisedtext@the\@edtext@level}}}{%

```

<sup>25</sup>This problem is caused by the way Xe<sub>La</sub>TeX manages right-to-left typesetting. David Carlisle explains it on <http://tex.stackexchange.com/a/333373/7712> and provides a potential solution, using `\vadjust`. However in some cases this adds spurious vertical spaces in `reledmac`. That is why we are using the solution explained below.



```

1409     {000}%
1410   }%
1411   {%
1412   \ifnumequal%
1413     {\xabslineref{edtext:start:\csuse{thisedtext@the\@edtext@level}}}%
1414     {\xabslineref{edtext:end:\csuse{thisedtext@the\@edtext@level}}}%
1415     {\flag@start}%
1416     {\flag@end}%
1417   }%
1418 }%
1419 %

```

**\flag@start@later** \flag@start@later and \flag@end@later: these send the \@ref@later to the line-list file command to the line-list file

**\flag@end@later**

```

1420 \newcommand*{\flag@start@later}{%
1421   \ifledRcol%
1422     \write\linenum@outR{\string\@ref@later []}%
1423   \else%
1424     \write\linenum@out{\string\@ref@later []}%
1425   \fi%
1426 }%
1427 \newcommand{\flag@end@later}{%
1428   \ifledRcol%
1429     \write\linenum@outR{[]}%
1430   \else%
1431     \write\linenum@out{[]}%
1432   \fi%
1433 }
1434 %

```

**\startsub** \startsub and \endsub turn sub-lineation on and off, by writing appropriate instructions to the line-list file. When sub-lineation is in effect, the line number counter is frozen and the sub-line counter advances instead. If one of these commands appears in the middle of a line, it 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 attached to the last word preceding the change, not the first word that follows the change. This is because sub-lineation will often turn on and off in mid-line—stage directions, for example, often are mixed with dialogue in that way—and when a line is mixed we want to label it using the system that was in effect at its start. But when sub-lineation begins at the very start of a line we have a problem, if we don't put in this code.

```

1435
1436
1437 \newcommand*{\startsub}{\dimen0\lastskip
1438   \ifdim\dimen0>0pt \unskip \fi
1439   \ifledRcol \write\linenum@outR{\string\sub@on}%

```

```

1440 \else      \write\linenum@out{\string\sub@on}%
1441 \fi
1442 \ifdim\dimen0>Opt \hskip\dimen0 \fi}
1443 \def\endsub{\dimen0\lastskip
1444 \ifdim\dimen0>Opt \unskip \fi
1445 \ifledRcol \write\linenum@outR{\string\sub@off}%
1446 \else      \write\linenum@out{\string\sub@off}%
1447 \fi
1448 \ifdim\dimen0>Opt \hskip\dimen0 \fi}
1449
1450 %

```

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

```

1451 \newcommand*{\advanceline}[1]{\leavevmode%
1452 \ifledRcol \write\linenum@outR{\string\@adv[#1]}%
1453 \else      \write\linenum@out{\string\@adv[#1]}%
1454 \fi}%
1455 }
1456 %

```

**\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.

```

1457
1458 \newcommand*{\setline}[1]{%
1459 \leavevmode%
1460 \ifnum#1<\z@
1461 \led@warn@BadSetline
1462 \else
1463 \ifledRcol \write\linenum@outR{\string\@set[#1]}%
1464 \else      \write\linenum@out{\string\@set[#1]}%
1465 \fi
1466 \fi}
1467
1468 %

```

**\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.

```

1469
1470 \newcommand*{\setlinenum}[1]{%
1471 \ifnum#1<\z@
1472 \led@warn@BadSetlinenum
1473 \else
1474 \ifledRcol \write\linenum@outR{\string\l@d@set[#1]}
1475 \else      \write\linenum@out{\string\l@d@set[#1]} \fi
1476 \fi}
1477
1478 %

```

`\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.

```

1479
1480 \newcommand*{\startlock}{%
1481   \ifledRcol \write\linenum@outR{\string\lock@on}%
1482   \else      \write\linenum@out{\string\lock@on}%
1483   \fi}
1484 \def\endlock{%
1485   \ifledRcol \write\linenum@outR{\string\lock@off}%
1486   \else      \write\linenum@out{\string\lock@off}%
1487   \fi}
1488 %

```

`\ifl@dskipnumber` In numbered text `\skipnumbering` will suspend the numbering for that particular line.

```

\ifl@dskipversenumber
\l@dskipnumbertrue
\l@dskipnumberfalse
\skipnumbering
1489 \newif\ifl@dskipnumber
1490 \newif\ifl@dskipversenumber%
1491 \newcommand*{\skipnumbering}{%
1492   \leavevmode%
1493   \ifledRcol%
1494     \ifinstanza%
1495       \write\linenum@outR{\string\n@num@stanza}%
1496     \else%
1497       \write\linenum@outR{\string\n@num}%
1498     \fi%
1499     \advanceline{-1}%
1500   \else%
1501     \ifinstanza%
1502       \write\linenum@out{\string\n@num@stanza}%
1503     \else%
1504       \write\linenum@out{\string\n@num}%
1505     \fi%
1506     \advanceline{-1}%
1507   \fi%
1508 }%
1509
1510 %

```

## 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. 145). 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 cannot 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.

```
1511 \list@create{\end@lemmas}
1512 %
```

`\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.

```
1513 \newcommand{\dummy@edtext}[2]{#1}
1514 %
```

`\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`.

```
1515 \newcommand{\dummy@edtext@showlemma}[2]{\showlemma{#1}}%
1516 %
```

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

`\no@expands` `\morenoexpands` 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>26</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— $\TeX$  seems to

<sup>26</sup>Since 'control sequences equivalent to characters are not expandable'—*The TeXbook*, answer to Exercise 20.14.

get confused about the difference between alignment parameters and macro parameters; those that use temporary control sequences; and those that look carefully at what the current font is.

(The `\copyright` macro defined in PLAIN  $\TeX$  has this sort of problem as well, but 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.

The `\new@series` command also adds `\let\footnote(X)\@gobble` to the end of the `\no@expands` macro for the series  $\langle X \rangle$ .

(A related problem, not addressed by these two macros, is that of characters whose category code are changed by any of 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 characters, using Lua $\TeX$  or Xe $\TeX$ .)

```

1517 \newcommand*{\no@expands}{%
1518   \let\select@lemmafnt=0%
1519   \let\startsub=\relax \let\endsub=\relax
1520   \let\startlock=\relax \let\endlock=\relax
1521   \let\edlabel=\@gobble
1522   \let\setline=\@gobble \let\advanceline=\@gobble
1523   \let\sameword\sameword@inedtext%
1524   \let\edtext=\dummy@edtext
1525   \let\edindex\dummy@edindex%
1526   \l@dtabnoexpands
1527   \morenoexpands}
1528 \let\morenoexpands=\relax
1529
1530 %

```

`\@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.

```

1531 \newcommand{\@tag}{}

```

```
1532 %
```

**\@edtext@level** This counter is increased by 1 at each level of `\edtext`.

```
1533 \newcount\@edtext@level%
1534 \@edtext@level=0%
1535 %
```

**\if@edtext@secondarg@** This boolean is set to TRUE before reading the second argument of a `\edtext`. It is tested on some macro which must be executed only inside a second argument.

```
1536 \newif\if@edtext@secondarg%
1537 %
```

**\theedtext** The `edtext` counter is increased at each `\edtext` command. It is used to add to insert hyperlinks between a notes and the lemma.

```
1538 \newcounter{edtext}
1539 \renewcommand{\theedtext}{edtxt@arabic{edtext}}%
1540 %
```

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

```
1541 \newcommand{\edtext}[2]{\leavevmode%
1542 %
```

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

```
1543 \ifnumberedpar%
1544 %
```

we increment the `\@edtext@level`  $\TeX$  counter to know in which level of `\edtext` we are.

```
1545 \global\advance\@edtext@level by 1%
1546 %
```

We also increase the `edtext`  $\TeX$  counter to insert a `hypertarget` if the `hyperref` package is loaded, and also works with `\edtext` on right-to-left typesetting with  $\XeTeX$ .

We store the value for the current level in a global macro. So we have one macro by level of `\edtext`. That is required, because `\edtext` can contain `\edtext`.

```
1547 \stepcounter{edtext}%
1548 \csxdef{thisedtext@the\@edtext@level}{\theedtext}%
1549 %
```

By default, we do not use `\lemma`

```
1550 \global\@lemmacommand@false%
1551 %
```

```

1552 \begingroup%
1553 %

```

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`.

```

1554 \ifledRcol%
1555 \ifcsvoid{sw@list@edtextR@the\@edtext@level}%
1556 {\global\let\sw@inthisedtext\empty}%
1557 {\expandafter\gl@p\csname sw@list@edtextR@the\@edtext@level\
endcsname\to\sw@inthisedtext}%
1558 \else%
1559 \ifcsvoid{sw@list@edtext@the\@edtext@level}%
1560 {\global\let\sw@inthisedtext\empty}%
1561 {\expandafter\gl@p\csname sw@list@edtext@the\@edtext@level\
endcsname\to\sw@inthisedtext}%
1562 \fi%
1563 %

```

`\@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.

```

1564 \global\renewcommand{\@tag}{%
1565 \no@expands #1%
1566 }%
1567 %

```

`\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`.

```

1568 \set@line%
1569 %

```

`\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`.

```

1570 \ifledRcol \global\insert@countR \z@%
1571 \else \global\insert@count \z@ \fi%
1572 %

```

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.



```

1573 \edtext@secondarg@true%
1574 \ignorespaces #2\relax%
1575 \edtext@secondarg@false%
1576 %

```

With  $\text{\LaTeX}$ , you must track whether the language reads left to right (English) or right to left (Arabic). `reledmac` defines an `\if@RTL` boolean test is not already defined.

```

1577 \if@RTL%
1578 \flag@start@RTL%
1579 \else%
1580 \flag@start%
1581 \fi%
1582 %

```

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

```

1583 \if@lemmacommand%
1584 \ifledRcol%
1585 \write\linenum@outR{\string\@lemma}%
1586 \else%
1587 \write\linenum@out{\string\@lemma}%
1588 \fi%
1589 \fi%
1590 %

```

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.

```

1591 \endgroup%
1592 \ifdef\hypertarget{%
1593 {%
1594 \Hy@raisedlink@left{\hypertarget{\csuse{thisedtext@the\@edtext@level}:start}{}}%
1595 \showlemma{#1}%
1596 \Hy@raisedlink{\hypertarget{\csuse{thisedtext@the\@edtext@level}:end}{}}%
1597 }%
1598 {%
1599 \showlemma{#1}%
1600 }%
1601 %

```

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.

```

1602 \ifx\end@lemmas\empty \else%
1603 \gl@p\end@lemmas\to\x@lemma%
1604 \x@lemma%
1605 \global\let\x@lemma=\relax%
1606 \fi%
1607 \if@RTL%
1608 \flag@end@RTL%
1609 \else%
1610 \flag@end%
1611 \fi%
1612 %

```

We switch some flags to false.

- 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`.

```

1613 \global\@noneed@Footnotefalse%
1614 \global\advance\@edtext@level by -1%
1615 \global\@lemmacommand@false%
1616 %

```

We also reset `\@beforeinsertofthisedtext`

```

1617 \global\let\@beforeinsertofthisedtext\relax%
1618 %

```

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

```

1619 \else%
1620 \showlemma{#1} (\textbf{\textsc{Edtext outside numbered paragraph}})\
led@err@edtextoutsidepstart%
1621 \fi%
1622 }%
1623
1624
1625 %

```

`\@beforeinsertofthisedtext` `\@beforeinsertofthisedtext` is an internal macro. `reledmac` or `reledpar` can add in this macro any content required to be executed before doing any `\insert` related to a `\edtext`. Its content is `\let` equal to `\relax` at the end of every `\edtext`.

```

1626 \let\@beforeinsertofthisedtext\relax
1627 %

```

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

```

1628 \newif\ifnumberline
1629 \numberlinetrue
1630 %

```

**\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.

```

1631 \newcommand*{\set@line}{%
1632   \ifl@edRcol
1633     \ifx\line@listR\empty
1634       \global\noteschanged@true
1635       \xdef\l@d@nums{000|000|000|000|000|000|\edfont@info}%
1636     \else
1637       \gl@p\line@listR\to\@tempb
1638       \xdef\l@d@nums{\@tempb|\edfont@info}%
1639       \global\let\@tempb=\undefined
1640     \fi
1641   \else
1642     \ifx\line@list\empty
1643       \global\noteschanged@true
1644       \xdef\l@d@nums{000|000|000|000|000|000|\edfont@info}%
1645     \else
1646       \gl@p\line@list\to\@tempb
1647       \xdef\l@d@nums{\@tempb|\edfont@info}%
1648       \global\let\@tempb=\undefined
1649     \fi
1650   \fi}
1651 %
1652 %

```

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

```

1653 \newcommand*{\edfont@info}{\f@encoding/\f@family/\f@series/\f@shape}
1654 %
1655 %

```

## 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. 128).

```

1656 \newcommand*{\lemma}[1]{%
1657   \global\@lemmacommand@true%
1658   \global\renewcommand{\@tag}{%
1659     \no@expands #1%
1660   }%
1661   \ignorespaces%
1662 }%
1663 %

```

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

```

1664 \newcommand{\@lemma}{%
1665   \booltrue{lemmacommand@the\@edtext@level}%
1666 }%
1667 %

```

**\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.

```

1668 \newif\if@lemmacommand@%
1669 %

```

### 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. 99): 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.

```

1670 \newcommand*{\linenum}[1]{%
1671   \xdef\@tempa{#1|}|}|}|}|}|noexpand\\l@d@nums}%
1672   \global\let\l@d@nums=\empty
1673   \expandafter\line@set\@tempa|\\ignorespaces}
1674 %

```

**\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.

```

1675 \def\line@set#1|#2\|#3|#4\|{%
1676   \gdef\@tempb{#1}%
1677   \ifx\@tempb\empty
1678     \l@d@add{#3}%
1679   \else
1680     \l@d@add{#1}%
1681   \fi
1682   \gdef\@tempb{#4}%
1683   \ifx\@tempb\empty\else
1684     \l@d@add{|\}\line@set#2\|#4\|}%
1685   \fi}
1686 %

```

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

```

1687 \newcommand{\l@d@add}[1]{\xdef\l@d@nums{\l@d@nums#1}}
1688
1689 %

```

## 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 `reledmac` 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 explains how, for same `\sameword`, absolute ranks are transformed to relative ranks.

```
At line 1:
absolute rank 1 becomes relative rank 1-0 = 1
1 is stored for this \sameword and line 1
At line 2:
absolute rank 2 becomes relative rank 2-1 = 1
absolute rank 3 becomes relative rank 3-1 = 2
3 is stored for this \sameword and line 2
At line 3:
no \sameword for this line.
3 is stored for this \sameword and line 3
At line 4:
absolute rank 4 becomes relative rank 4-3 = 1
4 is stored for this \sameword and line 4
```

2. Create lists of lists of `\sameword` by depth of `\edtext`. That is: create a list for `\edtexts` of level 1, a list for `\edtexts` of level 2, a list for `\edtexts` of level 3 etc. For each `\edtext` in these lists, we store all of the relative ranks of `\saweword` which are called as lemma information. That is: 1) either called in the first argument of `\sameword`, or, 2) 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 levels 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 its `\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 a list of relative ranks 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 a method of knowing the absolute line number associated to a `\sameword` is required because a `\edtext` can overlap many lines, but `\sameword` can't get it.
  - \* When reading the auxiliary file, we get the value associated to the pair composed by the current marked word and the current absolute line number. To this value, we subtract the value associated to the pair composed by the current marked word and the previous absolute line number. Let the result be  $n$ .
- If  $n > 1$ , that means the current word appears more than once in its line. In this case, we call `\showwordrank` with the word as the first argument and  $r$  as the 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 an 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 the dynamic name of macro with `\csname`.<sup>27</sup>

Because there is a bug with `\detokenize` and  $\text{\XeTeX}$  when using non BMP characters<sup>28</sup>, we detokenize only for non- $\text{\XeTeX}$  engines. In any case, in  $\text{\XeTeX}$  a `\csname` construction can contain UTF-8 characters without a problem, as UTF-8 characters are not managed with category codes, but instead read directly as UTF-8 characters.

```

1690 \newcommand{\get@sw@txt}[1]{%
1691   \begingroup%
1692     \let\sameword\@firstofone%Allow to have nested \sameword
1693     \ifxetex%
1694       \xdef\sw@txt{#1}%
1695     \else%
1696       \expandafter\xdef\expandafter\sw@txt\expandafter{\detokenize{#1}}%
1697     \fi%
1698   \endgroup%
1699 }%
1700 %

```

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

<sup>27</sup>See <http://tex.stackexchange.com/q/244538/7712>.

<sup>28</sup><http://sourceforge.net/p/xetex/bugs/108/>

```

1701 \newcommandx{\sameword}[2][1,usedefault]{%
1702     \leavevmode%
1703     \get@sw@txt{#2}%
1704 %

```

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

```

1705 \unless\ifledRcol%
1706     \csnumgdef{sw@\sw@txt}{\csuse{sw@\sw@txt}+\@ne}%
1707 %

```

Then, write its value to the numbered file.

```

1708     \protected@write\linenum@out{}\string\@sw{\sw@txt}{\csuse{sw@\sw@txt}
1709     }{#1}}%
1709 %

```

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

```

1710 \else%
1711     \csnumgdef{sw@\sw@txt}{\csuse{sw@\sw@txt}+\@ne}%
1712     \protected@write\linenum@outR{}\string\@sw{\sw@txt}{\csuse{sw@\sw@txt}
1713     }{#1}}%
1713 \fi%
1714 %

```

And print the word.

```

1715     #2%
1716 }%
1717 %

```

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

```

\if@addsw 18 \newif\if@addsw%
1719 %

```

**\@sw** The command printed in the auxiliary files.

```

1720 \newcommand{\@sw}[3]{%
1721     \get@sw@txt{#1}%
1722     \unless\ifledRcol%
1723 %

```

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

```

1724     \csxdef{sw@\sw@txt @\the\absline@num @\the\section@num}{#2}%
1725 %

```

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



```

1726 \numdef{\prev@line}{\the\absline@num-1}%
1727 \ifcsundef{sw@sw@txt @\prev@line @\the\section@num}{%
1728   \csnumgdef{sw@sw@txt @\prev@line @\the\section@num}{#2-1}%
1729   }{}%
1730 %

```

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

```

1731 \numdef{\the@sw}{#2-\csuse{sw@sw@txt @\prev@line @\the\section@num}}%
1732 %

```

And do the same thing for the right side.

```

1733 \else%
1734   \csxdef{sw@sw@txt @\the\absline@numR @\the\section@numR @R}{#2}%
1735   \numdef{\prev@line}{\the\absline@numR-1}%
1736   \ifcsundef{sw@sw@txt @\prev@line @\the\section@numR @R}{%
1737     \csnumgdef{sw@sw@txt @\prev@line @\the\section@numR @R}{#2-1}%
1738     }{}%
1739   \numdef{\the@sw}{#2-\csuse{sw@sw@txt @\prev@line @\the\section@numR @R}}%
1740   \fi%
1741 %

```

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

```

1742 \@tempcnta=\@edtext@level
1743 \@whilenum{\@tempcnta>0}\do{%
1744   \ifcsdef{sw@list@edtext@tmp@\the\@tempcnta}%
1745   {%
1746     \@addswfalse%
1747     \notbool{lemmacommand@\the\@tempcnta}%
1748     {\@addswtrue}%
1749     {\IfStrEq{#3}{inlemma}%
1750      {\@addswtrue}%
1751      {%
1752        \def\do##1{%
1753          \ifnumequal{##1}{\the\@tempcnta}%
1754          {\@addswtrue\listbreak}%
1755          }%
1756        }%
1757        \docsvlist{#3}%
1758      }%
1759    }%
1760    \if@addsw%
1761      \letcs{\@tmp}{sw@list@edtext@tmp@\the\@tempcnta}%
1762      \ifledRcol%
1763        \xright@appenditem{{\the@sw}{\the\absline@numR}}\to\@tmp%
1764      \else%
1765        \xright@appenditem{{\the@sw}{\the\absline@num}}\to\@tmp%
1766      \fi%
1767      \cslet{sw@list@edtext@tmp@\the\@tempcnta}{\@tmp}%

```

```

1768         \fi%
1769     }%
1770 }%
1771 \advance\@tempcnta by -1%
1772 }%
1773 }%
1774 %

```

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

```

1775 \newcommandx{\sameword@inedtext}[2][1,usedefault]{%
1776     \get@sw@txt{#2}%
1777     \unless\ifledRcol@%
1778 %

```

Just a precaution.

```

1779     \ifx\sw@list@inedtext\empty%
1780         \def\the@sw{999}%
1781         \def\this@absline{-99}%
1782     \else%
1783 %

```

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

```

1784         \glp\sw@list@inedtext\to\@tmp%
1785         \edef\the@sw{\expandafter\@firstoftwo\@tmp}%
1786         \edef\this@absline{\expandafter\@secondoftwo\@tmp}%
1787     \fi%
1788 %

```

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

```

1789     \ifcsdef{sw@\sw@txt @\this@absline @\the\section@num}{%
1790         \numdef{\prev@line}{\this@absline-1}%
1791         \numdef{\sw@atthisline}{\csuse{sw@\sw@txt @\this@absline @\the\
section@num}-\csuse{sw@\sw@txt @\prev@line @\the\section@num}}%
1792     }%
1793     {\numdef{\sw@atthisline}{0}}%
1794 %

```

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

```

1795     \ifnumgreater{\sw@atthisline}{1}%
1796         {\showwordrank{#2}{\the@sw}}%
1797         {#2}%
1798 %

```

And the same for right side.

```

1799     \else%
1800         \ifx\sw@list@inedtext\empty%

```

```

1801 \def\the@sw{999}%
1802 \def\this@absline{-99}%
1803 \else%
1804 \glp\sw@list@inedtext\to\@tmp%
1805 \edef\the@sw{\expandafter\@firstoftwo\@tmp}%
1806 \edef\this@absline{\expandafter\@secondoftwo\@tmp}%
1807 \fi%
1808 \ifcsdef{sw\sw@txt @\this@absline @\the\section@numR @R}{%
1809 \numdef{\prev@line}{\this@absline-1}%
1810 \numdef{\sw@atthisline}{\csuse{sw\sw@txt @\this@absline @\the\
section@numR @R}-\csuse{sw\sw@txt @\prev@line @\the\section@numR @R}}%
1811 }%
1812 {\numdef{\sw@atthisline}{0}}%
1813 \ifnumgreater{\sw@atthisline}{1}%
1814 {\showwordrank{#2}{\the@sw}}%
1815 {#2}%
1816 \fi%
1817 }%
1818 %

```

`\showwordrank` Finally, the way the rank will be printed.

```

1819 \newcommand{\showwordrank}[2]{%
1820 #1\textsuperscript{#2}%
1821 }%
1822 %

```

## 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`

`\raw@text` Here are numbers and flags that are used internally in the course of the paragraph decomposition.

`\ifnumberedpar@` 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.

`\numberedpar@true`

`\numberedpar@false`

`\num@lines`

`\one@line`

`\par@line`

```

1823 \newbox\raw@text
1824 \newif\ifnumberedpar@
1825 \newcount\num@lines
1826 \newbox\one@line
1827 \newcount\par@line
1828 %

```

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

```

\thepstart
\ifat@every@pstart@star@
1829 \newcommand{\AtStartEveryPstart}[1]{%
1830   \ifstrempy{#1}%
1831   {\gdef\@at@start@every@pstart{}}%
1832   {\gdef\@at@start@every@pstart{#1}}%
1833 }%
1834 \def\@at@start@every@pstart{}%
1835
1836 \newif\ifat@every@pstart@star@%
1837 \newcommand{\AtEveryPstart}[1]{%
1838   \ifstrempy{#1}%
1839   {\gdef\@at@every@pstart{}}%
1840   {\gdef\@at@every@pstart{\noindent#1}}%
1841   \global\@at@every@pstart@star@false%
1842 }%
1843 \WithSuffix\newcommand\AtEveryPstart*[1]{%
1844   \ifstrempy{#1}%
1845   {\gdef\@at@every@pstart{}}%
1846   {\gdef\@at@every@pstart{#1}}%
1847   \global\@at@every@pstart@star@true%
1848 }%
1849 \def\@at@every@pstart{}%
1850
1851 \newcounter{pstart}
1852 \renewcommand{\thepstart}{\bfseries\@arabic\c@pstart}. }
1853 \newif\ifnumberpstart
1854 \numberpstartfalse
1855 \newif\iflabelpstart
1856 \labelpstartfalse
1857 \newcommandx*\pstart[2][1,2,usedefault]{%
1858   \normal@pars%
1859   \ifboolexpr{%
1860     test {\ifstrempy{#1}}%
1861     and test {\ifstrempy{#2}}%
1862

```

```

1863 }%
1864   {\at@every@pstart}%
1865   {%
1866     \ifstrempy{#1}{-}{\noindent#1}%
1867     \ifstrempy{#2}{-}{#2}%
1868   }%
1869   \ifautopar%
1870     \autopar%
1871   \fi%
1872   \ifluatex%
1873     \edef\l@luatexttextdir@L{\the\textdir}%
1874   \fi%
1875   \@nbreaktrue%
1876   \ifnumbering \else%
1877     \led@err@PstartNotNumbered%
1878     \beginnumbering%
1879   \fi%
1880   \ifnumberedpar@%
1881     \led@err@PstartInPstart%
1882   \pend%
1883   \fi%
1884   \list@clear{\inserts@list}%
1885   \global\let\next@insert=\empty%
1886   \begingroup\normal@pars%
1887   \global\advance \l@dnumpstartsL\@ne
1888   \global\setbox\raw@text=\vbox\bgroup%
1889   \if@nbreak%
1890     \if@afterindent\else%
1891       \noindent%
1892       \global\@afterindenttrue%
1893     \fi%
1894   \fi%
1895   \ifautopar\else%
1896   \ifnumberpstart%
1897     \ifinstanza\else%
1898     \ifsidepstartnum\else%
1899       \thepstart%
1900     \fi%
1901   \fi%
1902   \fi%
1903   \fi%
1904   \numberedpar@true%
1905   \iflabelpstart\protected@edef\@currentlabel%
1906     {\p@pstart\thepstart}
1907   \fi%
1908   \l@dzeropenalties%
1909   \@at@start@every@pstart%
1910   \ignorespaces%because not automatically ignored if an optional argument
    is used (classical TeX behavior for space after commands)
1911 }

```

```
1912 %
```

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

```
1913 \newcommandx*{\pend}[2][1,2,usedefault]{\ifnumbering \else%
1914   \led@err@PendNotNumbered%
1915   \fi%
1916   \global\l@dskipversenumberfalse%
1917   \ifnumberedpar@ \else%
1918   \led@err@PendNoPstart%
1919   \fi%
1920 %
```

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.

```
1921 \l@dzeroopenalties%
1922 \@at@end@every@pend%
1923 \endgraf\global\num@lines=\prevgraf\egroup%
1924 \global\par@line=0%
1925 %
```

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`.

```
1926 \csnumdef{pstartline}{0}%
1927 \loop\ifvbox\raw@text%
1928   \csnumdef{pstartline}{\pstartline+\@ne}%
1929   \do@line%
1930   \ifbypstart@%
1931     \ifnumequal{pstartline}{1}{%
1932       \bgroup%
1933       \let\leavevmode\relax%
1934       \setline{1}%
1935       \egroup%
1936       \resetprevline@{}%
1937     }%
1938   \repeat%
1939 %
```

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

```
1940 \flush@notes%
1941 \endgroup%
1942 \ignorespaces%
1943 %
```

Increase pstart counter.

```

1944 \ifnumberpstart%
1945   \global\pstartnumtrue%
1946 \fi%
1947 \addtocounter{pstart}{1}%
1948 %

```

Print the optional arguments of \pend or the content printed after every \pend

```

1949 \normal@pars%
1950 \ifboolexpr{%
1951   test {\ifstrempy{#1}}%
1952   and test {\ifstrempy{#2}}%
1953 }%
1954   {\at@every@pend}%
1955   {%
1956   \ifstrempy{#1}{\noindent#1}%
1957   \ifstrempy{#2}{\#2}%
1958   }%
1959 %

```

Restore standard “nobreak” and “autopar” settings. Normally, \if@nobreak is true only immediately after a sectioning command (see latex.ltx file). As a \pstart... \pend structure can’t contain any sectioning command, we set \if@nobreak to false.

```

1960 \@nobreakfalse%
1961 \ifautopar%
1962   \autopar%
1963 \fi%
1964 }
1965 %

```

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%
\ifat@every@pend@star%
1969 \newif\ifat@every@pend@star%
1970 \newcommand{\AtEveryPend}[1]{%
1971   \ifstrempy{#1}%
1972   {\gdef\at@every@pend{}}%
1973   {\gdef\at@every@pend{\noindent#1}}%
1974   \global\at@every@pend@star@false%
1975 }%
1976 \WithSuffix\newcommand\AtEveryPend*[1]{%
1977   \ifstrempy{#1}%
1978   {\gdef\at@every@pend{}}%
1979   {\gdef\at@every@pend{#1}}%
1980   \global\at@every@pend@star@true%
1981 }%

```

```

1981 \xdef\at@every@pend{}%
1982
1983 %

```

**\AtEndEveryPend** Here a macro to insert automatically any content at the end of `\pend`, in numbered lines.

```

1984 \newcommand{\AtEndEveryPend}[1]{%
1985   \ifstrempy{#1}%
1986     {\xdef\@at@end@every@pend{}}%
1987     {\gdef\@at@end@every@pend{#1}}%
1988 }%
1989 \def\@at@end@every@pend{}%
1990 %

```

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

```

1991 \newcommand*\l@dzero penalties{%
1992   \brokenpenalty \z@ \clubpenalty \z@
1993   \displaywidowpenalty \z@ \interlinepenalty \z@ \predisplaypenalty \z@
1994   \postdisplaypenalty \z@ \widowpenalty \z@}
1995
1996 %

```

**\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.



```

1997 \newif\ifautopar
1998 \autoparfalse
1999 \newcommand*{\autopar}{
2000   \ifledRcol
2001     \ifnumberingR \else
2002     \led@err@AutoparNotNumbered
2003     \beginnumberingR
2004     \fi
2005   \else
2006     \ifnumbering \else
2007     \led@err@AutoparNotNumbered
2008     \beginnumbering
2009     \fi
2010   \fi
2011   \autopartrue
2012   \everypar{\setbox0=\lastbox
2013     \endgraf \vskip-\parskip
2014     \pstart \noindent \kern\wd0 \ifnumberpstart\ifinstanza\else\thepstart\
2015   \fi\fi
2016   \let\par=\pend}%
2017   \ignorespaces}
2018 %

```

**\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.

```

2018 \newcommand*{\normal@pars}{\everypar{} \let\par\endgraf}
2019
2020 %

```

**\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.

```

2021 \newif\ifautopar@pause
2022 %

```

## 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**

```

2023 \newcommand*{\l@dunhbox@line}[1]{\unhbox #1}
2024 \newcommand*{\do@line}{%
2025   {\vbadness=10000
2026     \splittopskip=\z@
2027     \do@linehook

```

```

2028 \l@demptyd@ta
2029   \global\setbox\one@line=\vsplit\raw@text to\baselineskip}%
2030   \unvbox\one@line \global\setbox\one@line=\lastbox
2031   \getline@num
2032   \IfStrEq{\led@pb@setting}{before}{\led@check@pb\led@check@nopb}{\}
2033   \ifnum\@lock>\@one
2034     \inserthangingsymboltrue
2035   \else
2036     \inserthangingsymbolfalse
2037   \fi
2038   \check@pb@in@verse
2039   \ifl@dhidenumber%
2040     \global\l@dhidenumberfalse%
2041     \f@x@l@cks%
2042   \else%
2043     \affixline@num%
2044   \fi%
2045 %

```

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

```

2046   \xifinlist{\the\l@dnumpstartsL}{\eled@sections@}%
2047     {\print@eledsection}%
2048     {\print@line}%
2049   \IfStrEq{\led@pb@setting}{after}{\led@check@pb\led@check@nopb}{\}
2050   }%
2051 %

```

### VII.2.2 Process for “normal” line

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

```

2052 \def\print@line{
2053 %

```

Insert the pstart number in side, if we are in the first line of a pstart.

```

2054   \affixpstart@num%
2055 %

```

The line will be boxed, to have the good width.

```

2056   \hb@xt@ \linewidth{
2057 %

```

User hook.

```

2058     \do@insidelinehook%
2059 %

```

Left line number

```
2060 \l@dld@ta%
2061 %
```

Prepare text to be inserted before notes.

```
2062 \if@firstlineofpage%
2063 \set@Xtxtbeforenotes%
2064 \global\@firstlineofpagefalse%
2065 \fi%
2066 %
```

Insert footnotes made of manuscripts data and critical footnotes.

```
2067 \ifdefstring{\ms@data@position}{msdata-regular}{%
2068 \insert@msdata%
2069 \add@inserts%
2070 \add@Xgroupbyline%
2071 }{%
2072 \add@inserts%
2073 \add@Xgroupbyline%
2074 \insert@msdata%
2075 }%
2076 %
```

Insert marginal notes.

```
2077 \affixside@note%
2078 %
```

Print left notes.

```
2079 \l@dlsn@te
2080 %
```

Boxes the line, writes information about new line in the numbered file.

```
2081 {\ledllfill\hb@xt@ \wd\one@line{\new@line%
2082 %
```

If we use Lua<sub>TeX</sub> then restore the direction.

```
2083 \ifluatex%
2084 \textdir\l@luatextextdir@L%
2085 \fi%
2086 %
```

Insert, if needed, the hanging symbol.

```
2087 \inserthangingsymbol%
2088 %
```

And so, print the line.

```
2089 \l@dunhbox@line{\one@line}}%
2090 %
```

Right line number

```

2091 \ledrlfill\l@drd@ta%
2092 %

```

Print right notes.

```

2093 \l@drsn@te
2094 }}%
2095 %

```

And reinsert penalties (for page breaking)...

```

2096 \add@penalties%
2097 }
2098 %

```

### 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.

```

2099 \def\print@eledsection{%
2100   \if@firstlineofpage%
2101     \set@Xtxtbeforenotes%
2102     \global\@firstlineofpagefalse%
2103   \fi%
2104   \ifdefstring{\ms@data@position}{msdata-regular}{%
2105     \insert@msdata%
2106     \add@inserts%
2107     \add@Xgroupbyline%
2108   }{%
2109     \add@inserts%
2110     \add@Xgroupbyline%
2111     \insert@msdata%
2112   }%
2113   \affixside@note%
2114   \numdef{\temp@}{\l@dnumstartsL-1}%
2115   \xifinlist{\temp@}{\eled@sections@}{\@nbreaktrue}{\@nbreakfalse}%
2116   \@eled@sectioningtrue%
2117   \csuse{eled@sectioning@the\l@dnumstartsL}%
2118   \@eled@sectioningfalse%
2119   \global\csundef{eled@sectioning@the\l@dnumstartsL}%
2120   \if@RTL%
2121     \hspace{-3\paperwidth}%
2122     {\hbox{\l@dunhbox@line{\one@line}} \new@line}%
2123   \else%
2124     \hspace{3\paperwidth}%
2125     {\new@line \hbox{\l@dunhbox@line{\one@line}}}%
2126   \fi%
2127   \vskip-\baselineskip%

```

```
2128 }
2129 %
```

### VII.2.4 Hooks

`\do@linehook` Two hooks into `\do@line`. The first is called at the beginning of `\do@line`, the second  
`\do@insidelinehook` is called in the line box. The second can, for example, have a `\markboth` command  
inside, the first can not.

```
2130 \newcommand*\do@linehook{}
2131 \newcommand*\do@insidelinehook{}
2132 %
```

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

```
2133 \newcommand*\dolinehook[1]{\gdef\do@linehook{#1}}%
2134 \newcommand*\doinsidelinehook[1]{\gdef\do@insidelinehook{#1}}%
2135
2136 %
```

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

`\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  
`\l@drd@ta` notes.

```
\l@dcsnotetext
2137 \newcommand*\l@emptyd@ta{}%
\l@dcsnotetext@l
2138 \gdef\l@dld@ta{}%
\l@dcsnotetext@r
2139 \gdef\l@drd@ta{}%
2140 \gdef\l@dcsnotetext@l{}%
2141 \gdef\l@dcsnotetext@r{}%
2142 \gdef\l@dcsnotetext{}%
2143
2144 %
```

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

```
\l@drsn@te
2145 \newcommand*\l@dlsn@te{}%
2146 \hb@xt@ \z@{\hss\box\l@dldp@rbox\kern\ledlsnotesep}}
2147 \newcommand*\l@drsn@te{}%
2148 \hb@xt@ \z@{\kern\ledrsnotesep\box\l@drp@rbox\hss}}
2149
2150 %
```

`\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`.

```

2151 \newcommand*{\ledllfill}{\hfil}
2152 \newcommand*{\ledrlfill}{}
2153
2154 %

```

## 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.

```

2155 \newcommand*{\getline@num}{%
2156   \global\advance\absline@num \@ne%
2157   \do@actions
2158   \do@ballast
2159   \ifnumberline
2160     \ifsublines@
2161       \ifnum\sub@lock<\tw@
2162         \global\advance\subline@num \@ne
2163       \fi
2164     \else
2165       \ifnum\@lock<\tw@
2166         \global\advance\line@num \@ne
2167         \global\subline@num \z@
2168       \fi
2169     \fi
2170   }
2171 }
2172 %

```

**\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,  $\TeX$  will be given extra encouragement to break the page here (see XI.2 p. 160).

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

**\c@ballast**

```

2173 \newcount\ballast@count
2174 \newcounter{ballast}
2175 \setcounter{ballast}{0}
2176 %

```

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.

```

2177 \newcommand*{\do@ballast}{\global\ballast@count \z@
2178 \beginingroup
2179 \advance\absline@num \@ne
2180 \ifnum\next@actionline=\absline@num
2181 \ifnum\next@action>-1001\relax
2182 \global\advance\ballast@count by -\c@ballast
2183 \fi
2184 \fi
2185 \endgroup}
2186 %

```

`\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.

`\do@actions@next`

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`.

```

2187 \newcommand*{\do@actions}{%
2188 \global\let\do@actions@next=\relax
2189 \ifnum\absline@num<\next@actionline\else
2190 %

```

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.

```

2191 \ifnum\next@action>-1001
2192 \global\page@num=\next@action
2193 \global\@firstlineofpagetrue%
2194 \ifbypage@
2195 \global\line@num=\z@ \global\subline@num=\z@
2196 \resetprevline@
2197 \fi
2198 \add@msdata@firstlineofpage%
2199 %

```

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`.)

```

2200 \else
2201 \ifnum\next@action<-4999
2202 \@l@tempcnta=-\next@action
2203 \advance\@l@tempcnta by -5001
2204 \ifsublines@
2205 \global\subline@num=\@l@tempcnta
2206 \else
2207 \global\line@num=\@l@tempcnta
2208 \fi
2209 %

```

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

```

2210     \else
2211         \@l@dttempcnta=-\next@action
2212         \advance\@l@dttempcnta by -1000
2213         \do@actions@fixedcode
2214     \fi
2215 \fi
2216 %

```

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.

```

2217     \ifx\actionlines@list\empty
2218         \gdef\next@actionline{1000000}%
2219     \else
2220         \gl@p\actionlines@list\to\next@actionline
2221         \gl@p\actions@list\to\next@action
2222         \global\let\do@actions@next=\do@actions
2223     \fi
2224 \fi
2225 %

```

Make the recursive call, if necessary.

```

2226 \do@actions@next}
2227
2228 %

```

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

```

2229 \newcommand*{\do@actions@fixedcode}{%
2230     \ifcase\@l@dttempcnta
2231     \or% % 1001 = starting sublineation
2232         \global\sublines@true
2233     \or% % 1002 = ending sublineation
2234         \global\sublines@false
2235     \or% % 1003 = starting locking number
2236         \global\@lock=\@ne
2237     \or% % 1004 = ending locking number
2238         \ifnum\@lock=\tw@
2239             \global\@lock=\thr@@
2240         \else
2241             \global\@lock=\z@
2242         \fi
2243     \or% % 1005 = starting locking subnumber
2244         \global\sub@lock=\@ne
2245     \or% % 1006 = ending locking subnumber
2246         \ifnum\sub@lock=\tw@
2247             \global\sub@lock=\thr@@
2248         \else

```



```

2249 \global\sub@lock=\z@
2250 \fi
2251 \or% % 1007 = skipping numbering
2252 \l@dskipnumbertrue
2253 \or% % 1008 = skipping numbering in stanza
2254 \l@dskipversenumbertrue%
2255 \or% % 1009 = hiding number
2256 \l@dhidenumbertrue
2257 \or% % 1010 = inserting msdata
2258 \add@msdata%
2259 \else
2260 \led@warn@BadAction
2261 \fi}
2262
2263
2264 %

```

## 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`  $\leq$  `\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.

```

2265 \newcommand*{\affixline@num}{%
2266 %

```

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).

```

2267 \ifl@groupnotesL@else
2268 \ifnumberline
2269 \ifl@dskipnumber
2270 \global\l@dskipnumberfalse
2271 \else
2272 \ifsublines@

```

```

2273 \l@dttempcntb=\subline@num
2274 \ifnum\subline@num>\c@firstsublinenum
2275 \l@dttempcnta=\subline@num
2276 \advance\l@dttempcnta by-\c@firstsublinenum
2277 \divide\l@dttempcnta by\c@sublinenumincrement
2278 \multiply\l@dttempcnta by\c@sublinenumincrement
2279 \advance\l@dttempcnta by\c@firstsublinenum
2280 \else
2281 \l@dttempcnta=\c@firstsublinenum
2282 \fi
2283 %

```

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.

```

2284 \ch@cksub@l@ck
2285 %

```

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

```

2286 \else
2287 \l@dttempcntb=\line@num
2288 %

```

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

```

2289 \ifx\linenumberlist\empty
2290 \ifnum\line@num>\c@firstlinenum
2291 \l@dttempcnta=\line@num
2292 \advance\l@dttempcnta by-\c@firstlinenum
2293 \divide\l@dttempcnta by\c@linenumincrement
2294 \multiply\l@dttempcnta by\c@linenumincrement
2295 \advance\l@dttempcnta by\c@firstlinenum
2296 \else
2297 \l@dttempcnta=\c@firstlinenum
2298 \fi
2299 \else
2300 %

```

The `\linenumberlist` was not `\empty`, so here is Wayne's numbering mechanism. This takes place in  $\TeX$ 's mouth.

```

2301 \l@dttempcnta=\line@num
2302 \edef\rem@inder{\,\linenumberlist,\number\line@num,}%
2303 \edef\sc@n@list{\def\noexpand\sc@n@list
2304 {###1,\number\l@dttempcnta,###2|{\def\noexpand\rem@inder
{###2}}}%
2305 \sc@n@list\expandafter\sc@n@list\rem@inder|}%
2306 \ifx\rem@inder\empty%
2307 \advance\l@dttempcnta\@ne
2308 \fi
2309 \fi
2310 %

```

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

```
2311 \ch@ck@l@ck
2312 \fi
2313 %
```

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

```
2314 \ifnum\@l@tempcnta=\@l@tempcntb
2315 \ifl@dskipversenumber\else
2316 %
```

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  $\text{\LaTeX}$  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
2317 \if@twocolumn
2318 \if@firstcolumn
2319 \gdef\l@dld@ta{\llap{\leftlinenum}}%
2320 \else
2321 \gdef\l@drd@ta{\rlap{\rightlinenum}}%
2322 \fi
2323 \else
2324 \ifboolexpr{bool {l@dprintingcolumns} and test {\
ifnumgreater{\line@margin@columns}{\m@ne}}}%
2325 {\@l@tempcntb=\line@margin@columns}%
2326 {\@l@tempcntb=\line@margin}%
2327 \ifnum\@l@tempcntb>\@ne
2328 \advance\@l@tempcntb \page@num
2329 \fi
2330 \ifodd\@l@tempcntb
2331 \gdef\l@drd@ta{\rlap{\rightlinenum}}%
2332 \else
2333 \gdef\l@dld@ta{\llap{\leftlinenum}}%
2334 \fi
2335 \fi
2336 \fi
2337 \fi
2338 %
```

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

```

2339     \f@x@l@cks
2340     \fi
2341   \fi
2342 \fi
2343 }
2344
2345 %

```

`\ch@cksub@l@ck` These macros handle line number locking for `\affixline@num`. `\ch@cksub@l@ck` checks subline locking. If it fails, then we disable the line-number display by setting the counters to arbitrary but unequal values.

```

2346 \newcommand*{\ch@cksub@l@ck}{%
2347   \ifcase\sub@lock
2348     \or
2349     \ifnum\sublock@disp=\@ne
2350       \@l@tempcntb=\z@ \@l@tempcnta=\@ne
2351     \fi
2352     \or
2353     \ifnum\sublock@disp=\tw@ \else
2354       \@l@tempcntb=\z@ \@l@tempcnta=\@ne
2355     \fi
2356     \or
2357     \ifnum\sublock@disp=\z@
2358       \@l@tempcntb=\z@ \@l@tempcnta=\@ne
2359     \fi
2360   \fi}
2361 %

```

Similarly for line numbers.

```

2362 \newcommand*{\ch@ck@l@ck}{%
2363   \ifcase\@lock
2364     \or
2365     \ifnum\lock@disp=\@ne
2366       \@l@tempcntb=\z@ \@l@tempcnta=\@ne
2367     \fi
2368     \or
2369     \ifnum\lock@disp=\tw@ \else
2370       \@l@tempcntb=\z@ \@l@tempcnta=\@ne
2371     \fi
2372     \or
2373     \ifnum\lock@disp=\z@
2374       \@l@tempcntb=\z@ \@l@tempcnta=\@ne
2375     \fi
2376   \fi}
2377 %

```

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

```

2378 \newcommand*{\f@x@l@cks}{%
2379   \ifcase\@lock
2380   \or
2381     \global\@lock=\tw@
2382   \or \or
2383     \global\@lock=\z@
2384   \fi
2385   \ifcase\sub@lock
2386   \or
2387     \global\sub@lock=\tw@
2388   \or \or
2389     \global\sub@lock=\z@
2390   \fi}
2391
2392 %

```

## 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 \rightpstartnum commands. After the try, it is set to FALSE.

```

\leftpstartnum93
\rightpstartnum94 \newif\ifsidepstartnum
\ifsidepstartnum95 \newcommand*{\affixpstart@num}{%
2396   \ifsidepstartnum
2397     \if@twocolumn
2398       \if@firstcolumn
2399         \gdef\l@dld@ta{\llap{\leftpstartnum}}}%
2400       \else
2401         \gdef\l@drd@ta{\rlap{\rightpstartnum}}}%
2402       \fi
2403     \else
2404       \@l@tempcntb=\line@margin%
2405       \ifnum\@l@tempcntb>\@ne
2406         \advance\@l@tempcntb \page@num
2407       \fi
2408       \ifodd\@l@tempcntb
2409         \gdef\l@drd@ta{\rlap{\rightpstartnum}}}%
2410       \else

```

```

2411 \gdef\l@dld@ta{\llap{{\leftstartnum}}}%
2412 \fi
2413 \fi
2414 \fi
2415 }
2416 %
2417
2418
2419 \newif\ifpstartnum
2420 \pstartnumtrue
2421 \newcommand*{\leftpstartnum}{
2422   \ifpstartnum\thepstart
2423   \kern\linenumsep\fi
2424   \global\pstartnumfalse
2425 }
2426 \newcommand*{\rightpstartnum}{
2427   \ifpstartnum
2428   \kern\linenumsep
2429   \thepstart
2430   \fi
2431   \global\pstartnumfalse
2432 }
2433 %

```

## XI Restoring footnotes and penalties

Because of the paragraph decomposition process in order to number line, `reledmac` must hack the standard way  $\TeX$  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  $\TeX$  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.

```

2434 \list@create{\inserts@list}
2435 %

```

`\add@inserts` `\add@inserts` is the penultimate macro used by `\do@line`; it takes insertions saved in a list macro and sends them onto the vertical list.

It may call itself recursively, and to do this efficiently (using  $\TeX$ 's optimization for tail recursion), we define a control-sequence called `\add@inserts@next` that is always

the last thing that `\add@inserts` does. If there could be more inserts to process for this line, `\add@inserts@next` is set equal to `\add@inserts`; otherwise it is just `\relax`.

```
2436 \newcommand*{\add@inserts}{%
2437   \global\let\add@inserts@next=\relax
2438   %
```

If `\inserts@list` is empty, there are not any more notes or insertions for this paragraph, and we need not waste our time.

```
2439   \ifx\inserts@list\empty \else
2440   %
```

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.

```
2441   \ifx\next@insert\empty
2442     \ifx\insertlines@list\empty
2443       \global\noteschanged@true
2444       \gdef\next@insert{100000}%
2445     \else
2446       \gl@p\insertlines@list\to\next@insert
2447     \fi
2448   \fi
2449   %
```

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.

```
2450   \ifnum\next@insert=\absline@num
2451     \gl@p\inserts@list\to\@insert
2452     \@insert
2453     \global\let\@insert=\undefined
2454     \global\let\next@insert=\empty
2455     \global\let\add@inserts@next=\add@inserts
2456   \fi
2457 \fi
2458 %
```

Make the recursive call, if necessary.

```
2459 \add@inserts@next}
2460
2461 %
```

**`\add@Xgroupbyline`** If you use `\Xgroupbyline`, the insertion of the critical footnotes are not made immediately in `\add@inserts`, but the content to be inserted is stored, to be inserted in one block. This insertion in one block is made by `\add@Xgroupbyline`.

```
2462 \newcommand{\add@Xgroupbyline}{%
2463   \unless\ifnocritical%
```

```

2464 \def\do##1{%Looping on the series
2465 \def\do####1{%Looping on the ##1@forinserting command
2466 \ifcsdef{##1@forinserting@####1}{%
2467 \X@beforeinsertion{##1}%
2468 \if@ledgroup%
2469 \global\setbox\@nameuse{mp##1footins}=\vbox%
2470 \else%
2471 \insert\csname ##1footins\endcsname%
2472 \fi%
2473 {%
2474 \ifcsdef{Xhsize\csuse{series@display##1}@##1}%
2475 {\hsize \csuse{Xhsize\csuse{series@display##1}@##1}}%
2476 }%
2477 \if@ledgroup%
2478 \unvbox\@nameuse{mp##1footins}%
2479 \fi%
2480 \X@atbegininsertion{##1}%
2481 \ifcsstring{series@display##1}%
2482 {%
2483 \Xledsetnormalparstuff{##1}%
2484 \rule\z@\splittopskip%
2485 }%
2486 }%
2487 \csuse{##1@forinserting@####1}%
2488 \strut\par%
2489 }%
2490 \global\csundef{##1@forinserting@####1}%
2491 }%
2492 }%
2493 }%
2494 \ifcsdef{##1@forinserting}{%
2495 \dolistcslloop{##1@forinserting}%
2496 }{%
2497 \global\csundef{##1@forinserting}%
2498 }%
2499 \dolistloop{\@series}%
2500 \fi%
2501 }%
2502
2503
2504 %

```

## 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. 150). Finally, the penalty is checked to see that it does not go below  $-10000$ .

```

2505 \newcommand*{\add@penalties}{\@l@tempcnta=\ballast@count
2506 \ifnum\num@lines>\@ne
2507   \global\advance\par@line \@ne
2508   \ifnum\par@line=\@ne
2509     \advance\@l@tempcnta \clubpenalty
2510   \fi
2511   \@l@tempcntb=\par@line \advance\@l@tempcntb \@ne
2512   \ifnum\@l@tempcntb=num@lines
2513     \advance\@l@tempcnta \widowpenalty
2514   \fi
2515   \ifnum\par@line<\num@lines
2516     \advance\@l@tempcnta \interlinepenalty
2517   \fi
2518 \fi
2519 \ifnum\@l@tempcnta=\z@
2520   \relax
2521 \else
2522   \ifnum\@l@tempcnta>-10000
2523     \penalty\@l@tempcnta
2524   \else
2525     \penalty -10000
2526   \fi
2527 \fi}
2528
2529 %

```

### 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  $\text{\TeX}$ , 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.

```

2530 \newcommand*{\flush@notes}{%
2531 \xloop
2532 \ifx\inserts@list\empty \else
2533   \gl@p\inserts@list\to\@insert
2534   \@insert
2535   \global\let\@insert=\undefined

```

```

2536 \repeat}
2537
2538 %

```

**\@xloop** \@xloop is a variant of the PLAIN T<sub>E</sub>X \loop macro, useful when it's hard to construct a positive test using the T<sub>E</sub>X \if commands—as in \flush@notes above. One types \@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 T<sub>E</sub>X \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.

```

2539 \def\@xloop#1\repeat{%
2540   \def\body{#1\expandafter\body\fi}%
2541   \body}
2542
2543 %

```

## XI.4 Text before notes

**\set@Xtxtbeforenotes** The \set@Xtxtbeforenotes macro resets the Xtxtbeforenotes@⟨series⟩@typeset boolean to false. Just before the first note of the ⟨series⟩ in a page, the Xtxtbeforenotes will be inserted.

```

2544 \newcommand{\set@Xtxtbeforenotes}{%
2545   \unless\ifnocritical@%
2546     \def\do##1{%
2547       \global\togglefalse{Xtxtbeforenotes@##1@typeset}%
2548     }%
2549     \dolistloop{\@series}%
2550   \fi%
2551 }%
2552 %

```

**\insert@Xtxtbeforenotes** \insert@Xtxtbeforenotes{⟨series⟩}, called when inserting a note, will insert the text before the note if it is not already inserted. For paragraphed footnotes, it will insert it as a component of the first footnote. For other types of footnotes, it will insert it as a regular footnote.

```

2553 \newcommand{\insert@Xtxtbeforenotes}[1]{%
2554   \nottoggle{Xtxtbeforenotes@#1@typeset}{%
2555     \global\toggletrue{Xtxtbeforenotes@#1@typeset}%
2556     \ifcvoid{Xtxtbeforenotes@#1}{%
2557       \ifcsstring{series@display#1}{paragraph}%
2558       {\noindent\cuse{Xtxtbeforenotes@#1}}%
2559       {\expandafter\insert\csname#1footins\endcsname%
2560        \bgroup%
2561        \noindent%

```

```

2562         \csuse{\csuse{series@display#1}@begin@insert}{#1}%
2563         \strut\csuse{Xnotefontsize@#1}\csuse{Xtxtbeforenotes@#1}%
2564         \egroup%
2565     }%
2566 }%
2567 }%
2568 {}%
2569 }%
2570 %

```

## XII Critical footnotes

The footnote macros are adapted from those in PLAIN T<sub>E</sub>X, 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.

```

2571 \def\select@lemmafont#1|#2|#3|#4|#5|#6|#7|{\select@lemmafont#7|}
2572 \def\select@@lemmafont#1/#2/#3/#4|{%
2573     {\fontencoding{#1}\fontfamily{#2}\fontseries{#3}\fontshape{#4}%
2574     \selectfont}
2575
2576 %

```

### 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.

```

2577 \newcommand*{\footnoteoptions@}[3]{%
2578     \def\do##1{%
2579         \ifstrequal{#1}{L}{% In Leftside
2580             \xright@appenditem{\noexpand\setkeys[mac]{#3footnoteoption}{\
unexpanded{##1}}}{\to\inserts@list%
2581             \global\advance\insert@count \@ne% Increment the left insert
counter.

```

```

2582     }%
2583     {%
2584         \xright@appenditem{\noexpand\setkeys[mac]{#3footnoteoption}{\
unexpanded{##1}}}\to\inserts@listR%
2585         \global\advance\insert@countR \@ne% Increment the right insert
counter insert.
2586     }%
2587 }%
2588 \notblank{#2}{\docsvlist{#2}}{}% Parsing all options
2589 }
2590 %

```

### XII.3 Notes language

`\footnotelang@lua` `\footnotelang@lua` is called to remember the information about the direction of a lemma when Lua<sup>®</sup>TeX is used.

```

2591 \newcommand*{\footnotelang@lua}[1][1=L,usedefault]{%
2592     \ifstrequal{#1}{L}{%
2593         \xright@appenditem{\csxdef{footnote@luatextextdir}{\the\textdir}}\to\
inserts@list%Know the dir of lemma
2594         \global\advance\insert@count \@ne%
2595         \xright@appenditem{\csxdef{footnote@luatexpardir}{\the\pardir}}\to\
inserts@list%Know the dir of lemma
2596         \global\advance\insert@count \@ne%
2597     }%
2598     {%
2599         \xright@appenditem{\csxdef{footnote@luatextextdir}{\the\textdir}}\to\
inserts@listR%Know the dir of lemma
2600         \global\advance\insert@countR \@ne%
2601         \xright@appenditem{\csxdef{footnote@luatexpardir}{\the\pardir}}\to\
inserts@listR%Know the dir of lemma
2602         \global\advance\insert@countR \@ne%
2603     }%
2604 }
2605 %

```

`\footnotelang@poly` `\footnotelang@poly` is called to remember the information about the language of a lemma when polyglossia is used.

```

2606 \newcommand*{\footnotelang@poly}[1][1=L,usedefault]{%
2607     \ifstrequal{#1}{L}{%
2608         \if@RTL%
2609             \xright@appenditem{\csxdef{footnote@dir}{@RTLtrue}}\to\
inserts@list%Know the language used in the lemma
2610             \global\advance\insert@count \@ne%
2611         \else
2612             \xright@appenditem{\csxdef{footnote@dir}{@RTLfalse}}\to\
inserts@list%Know the language of lemma

```

```

2613 \global\advance\insert@count \@ne%
2614 \fi%
2615 \xright@appenditem{{\csxdef{footnote@lang}{\expandonce\language}}}\
to\inserts@list%Know the language of lemma
2616 \global\advance\insert@count \@ne%
2617 }%
2618 {%
2619 \if@RTL
2620 \xright@appenditem{{\csxdef{footnote@dir}{@RTLtrue}}}\to\
inserts@listR%Know the language of lemma
2621 \global\advance\insert@countR \@ne%
2622 \else
2623 \xright@appenditem{{\csxdef{footnote@dir}{@RTLfalse}}}\to\
inserts@listR%Know the language of lemma
2624 \global\advance\insert@countR \@ne%
2625 \fi
2626 \xright@appenditem{{\csxdef{footnote@lang}{\expandonce\language}}}\
to\inserts@listR%Know the language of lemma
2627 \global\advance\insert@countR \@ne%
2628 }%
2629 }
2630 %

```

## 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`.

```

2631 \newcommand*{\footplitskips}{%
2632   \interlinepenalty=\interfootnotelinepenalty
2633   \unless\ifl@dprintingpages%
2634     \floatingpenalty=\@MM%
2635   \fi%
2636   \splittopskip=\ht\strutbox \splitmaxdepth=\dp\strutbox
2637   \leftskip=\z@skip \rightskip=\z@skip}
2638
2639 %

```

`\normalfootnoterule` `\normalfootnoterule` is a standard footnote-rule macro, for use by a `footstart` macro: just the same as the PLAIN  $\TeX$  footnote rule.

```

2640 \let\normalfootnoterule=\footnoterule
2641 %

```

## XII.6 Footnotes arrangement

### XII.6.1 User level macro

`\Xarrangement` `\Xarrangement[ $\langle s \rangle$ ]{ $\langle arrangement \rangle$ }` The command calls, for each series, a specific command which set many counters and commands in order to define specific arrangement.

```

2642 \newcommandx{\Xarrangement}[2][1,usedefault]{%
2643   \def\do##1{%
2644     \csname Xarrangement@#2\endcsname{##1}%
2645   }%
2646   \ifstrempy{#1}%
2647     {%
2648       \dolistloop{\@series}%
2649     }%
2650   {
2651     \docsvlist{#1}%
2652   }%
2653 }%
2654 %

```

### 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 insert 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\vsiz
\let\Afootnote=\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.

```
2655 \newcommand*{\Xarrangement@normal}[1]{%
2656   \csdef{series@display#1}{normal}
2657   \expandafter\let\csname #1footstart\endcsname=\normalfootstart
2658   \expandafter\let\csname v#1footnote\endcsname=\normalvfootnote
2659   \expandafter\let\csname #1footfmt\endcsname=\normalfootfmt
2660   \expandafter\let\csname #1footgroup\endcsname=\normalfootgroup
2661   \expandafter\let\csname #1footnoterule\endcsname=%
2662                                     \normalfootnoterule
2663   \count\csname #1footins\endcsname=1000
2664   \dimen\csname #1footins\endcsname=\csuse{Xmaxhnotes@#1}
2665   \skip\csname #1footins\endcsname=\csuse{Xbeforenotes@#1}%
2666   \advance\skip\csname #1footins\endcsname by\csuse{Xafterrule@#1}%
2667   %
```

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  $\text{\TeX}$ ’s inserts.

```
2668   \csxdef{default@#1footins}{1000}%Use this to confine the notes to one
2669   side only
2670   %
```

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).

```
2670 \ifnoledgroup@else%
2671   \expandafter\let\csname mpv#1footnote\endcsname=\mpnormalvfootnote
2672   \expandafter\let\csname mp#1footgroup\endcsname=\mpnormalfootgroup
2673   \count\csname mp#1footins\endcsname=1000
```

```

2674 \dimen\csname mp#1footins\endcsname=\csuse{Xmaxhnotes@#1}
2675 \skip\csname mp#1footins\endcsname=\csuse{Xbeforenotes@#1}%
2676 \advance\skip\csname mp#1footins\endcsname by\csuse{Xafterrule@#1}%
2677 \fi
2678 }
2679
2680 %

```

**\normalvfootnote** We now begin a series of commands that do ‘normal’ footnote formatting: a format much like that implemented in PLAIN T<sub>E</sub>X, 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.

```

2681 \notbool{parapparatus@}{\newcommand*}{\newcommand}{\normalvfootnote}[2]{%
2682 \iftoggle{Xgroupbyline@#1}{%In the case we use \Xgroupbyline, the
insertion is done later, in \add@Xgroupbyline.
2683 \prepare@Xgroupbyline{#1}{#2}{\normalvfootnote@inserted}%
2684 }{%In the case we don't use \Xgroupbyline, the insertion is made directly
2685 \X@beforeinsertion{#1}%
2686 \insert\csname #1footins\endcsname{%
2687 \X@atbegininsertion{#1}%
2688 \normalvfootnote@inserted{#1}{#2}%
2689 }%
2690 }%
2691 }%
2692 %

```

**\normalvfootnote@inserted** The \normalvfootnote@inserted macro is expanded to the content to be add to a \insert for normal critical footnote.

```

2693 \notbool{parapparatus@}{\newcommand*}{\newcommand}{\normalvfootnote@inserted}[2]{%
2694 \nottoggle{Xgroupbyline@#1}{\noindent}{\csuse{Xhooknote@#1}%
2695 \csuse{Xnotefontsize@#1}%
2696 \footsplitskips
2697 \ifl@dpairing\ifl@dpaging\else%
2698 \setXnoteswidthliketwocolumns@{#1}%
2699 \fi\fi%
2700 \setXnotespositionliketwocolumns@{#1}%
2701 \spaceskip=\z@skip \xspaceskip=\z@skip%
2702 \csname #1footfmt\endcsname #2{#1}%
2703 }%
2704 %

```

```

\X@beforeinsertion\newcommand{\X@beforeinsertion}[1]{%
2706 \if@ledgroup\else%
2707 \insert@Xtxtbeforenotes{#1}%

```



```

2708 \fi%
2709 \csuse{Xbeforeinserting@#1}%
2710 }%
2711 %

```

```

\X@atbegininsertion12 \newcommand{\X@atbegininsertion}[1]{%
2713 \hspace=\expandafter\dimexpr\csuse{Xwidth@#1}\relax%
2714 }%
2715 %

```

And somewhat different versions of `\normalvfootnote` and `\normalvfootnote@inserted` for minipages.

```

\mpnormalvfootnote16 \notbool{parapparatus@}\newcommand*{\newcommand}{\mpnormalvfootnote}[2]{%
2717 \iftoggle{Xgroupbyline@#1}{%
2718 \prepare@Xgroupbyline{#1}{#2}{\mpnormalvfootnote@inserted}%
2719 }%
2720 {%
2721 \global\setbox\@nameuse{mp#1footins}%
2722 \vbox{%
2723 \unvbox\@nameuse{mp#1footins}%
2724 \mpnormalvfootnote@inserted{#1}{#2}%
2725 }%
2726 }%
2727 }%
2728 %
2729 %

```

```

\mpnormalvfootnote@inserted30 \newcommand{\mpnormalvfootnote@inserted}[2]{%
2731 \noindent\csuse{Xbhooknote@#1}%
2732 \csuse{Xnotefontsize@#1}%
2733 \hspace\columnwidth%
2734 \@parboxrestore%
2735 \color@begingroup%
2736 \csname #1footfmt\endcsname #2{#1}\color@endgroup%
2737 }%
2738 %

```

`\normalfootfmt` `\normalfootfmt` is a ‘normal’ macro to take the footnote line and page number information (see V.9 p. 99), 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.

```

2739 \notbool{parapparatus@}{\newcommand*{\newcommand}{\normalfootfmt}[4]{%
2740   \xstorelineinfo{#1}{#4}%
2741   \nottoggle{Xgroupbyline@#4}{\xledsetnormalparstuff{#4}}{}%
2742   \hangindent=\csuse{Xhangindent@#4}%
2743   \everypar{\hangindent=\csuse{Xhangindent@#4}}%
2744   \nottoggle{Xgroupbyline@#4}{\rule{z@}{\splittopskip}}{}%
2745   {\printlinefootnote{#1}{#4}}%
2746   \print@lemma{#1}{#2}{#4}%
2747   \csuse{Xwrapcontent@#4}{#3}%
2748   \nottoggle{Xgroupbyline@#4}{\strut\par}}{}%
2749 }%
2750 %

```

**\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 \Xprenotes@ 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.

```

2751 \newcommand*{\normalfootstart}[1]{%
2752 %

```

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. 220. Here is part of this algorithm, when the block of notes are ready to be printed.

```

2753   \ifdimequal{0pt}{\Xprenotes@}{}%
2754   {%
2755     \iftoggle{Xprenotes@}{%
2756       \togglefalse{Xprenotes@}%
2757       \skip\csname #1footins\endcsname=%
2758       \glueexpr\csuse{Xprenotes@}+\csuse{Xafterrule@#1}\relax%
2759     }%
2760   }%
2761   }%
2762   \vskip\skip\csname #1footins\endcsname%
2763 %

```

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. 218 for the general description of the problem.

```

2764 \leftskipOpt \rightskipOpt
2765 \ifl@dpairing\else%
2766   \hsize=\old@hsize%
2767 \fi%
2768 \setXnoteswidthliketwocolumns@{#1}%
2769 \setXnotespositionliketwocolumns@{#1}%
2770 %

```

And now, print the footnote's rule to finish the footnote's introduction.

```

2771 \print@Xfootnoterule{#1}%
2772 }%
2773 %

```

`\normalfootgroup` `\normalfootgroup` is a standard footnote-grouping macro: it sends the contents of the footnote-insert box to the output page without alteration.

```

2774 \newcommand*{\normalfootgroup}[1]{%
2775   \csuse{Xbhookgroup@#1}%
2776   \unvbox\csname #1footins\endcsname%
2777   \hsize=\old@hsize%
2778 }%
2779
2780 %

```

`\mpnormalfootgroup` A somewhat different version for minipages. Note that, in this case, we do not make distinctions between the `\Xfootgroup` and `\Xfootstarts` macros.

```

2781 \unless\ifnoledgroup@
2782 \newcommand*{\mpnormalfootgroup}[1]{%
2783   \vskip\skip\@nameuse{mp#1footins}
2784   \ifl@dpairing\ifparledgroup%
2785     \leavevmode\marks\parledgroup@{begin}%
2786     \marks\parledgroup@series{#1}%
2787     \marks\parledgroup@type{Xfootnote}%
2788   \fi\fi\normalcolor%
2789   \ifparledgroup%
2790     \ifl@dpairing%
2791     \else%
2792       \setXnoteswidthliketwocolumns@{#1}%
2793       \setXnotespositionliketwocolumns@{#1}%
2794       \print@Xfootnoterule{#1}%%
2795     \fi%
2796   \else%
2797     \setXnoteswidthliketwocolumns@{#1}%
2798     \setXnotespositionliketwocolumns@{#1}%
2799     \print@Xfootnoterule{#1}%%

```

```

2800 \fi%
2801 \setlength{\parindent}{Opt}
2802 \csuse{Xbhookgroup@#1}%
2803 \unvbox\csname mp#1footins\endcsname}}
2804 \fi
2805 %

```

### XII.6.3 Paragraphed footnotes

The paragraphed-footnote option reformats all the footnotes of one series for a page into a single paragraph; this is especially appropriate when the notes are numerous and brief. The code is based on *The TeXbook*, pp. 398–400, with alterations for our environment. This algorithm uses a considerable amount of save-stack space: a T<sub>E</sub>X 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 paragraphed; 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 paragraphed 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 paragraphed.

```

2806 \newcommand*{\Xarrangement@paragraph}[1]{%
2807   \csgdef{series@display#1}{paragraph}
2808   \expandafter\let\csname #1footstart\endcsname=\parafootstart
2809   \expandafter\let\csname v#1footnote\endcsname=\paravfootnote
2810   \expandafter\let\csname #1footfmt\endcsname=\parafootfmt
2811   \expandafter\let\csname #1footgroup\endcsname=\parafootgroup
2812   \count\csname #1footins\endcsname=1000
2813   \csxdef{default@#1footins}{1000}%Use this to confine the notes to one
side only
2814   \dimen\csname #1footins\endcsname=\csuse{Xmaxhnotes@#1}
2815   \skip\csname #1footins\endcsname=\csuse{Xbeforenotes@#1}%
2816   \advance\skip\csname #1footins\endcsname by\csuse{Xafterrule@#1}%
2817   \para@footsetup{#1}
2818 %

```

And the extra setup for minipages.

```

2819 \ifnoledgroup@else
2820   \expandafter\let\csname mpv#1footnote\endcsname=\mpparavfootnote
2821   \expandafter\let\csname mp#1footgroup\endcsname=\mpparafootgroup
2822   \count\csname mp#1footins\endcsname=1000
2823   \dimen\csname mp#1footins\endcsname=\csuse{Xmaxhnotes@#1}
2824   \skip\csname mp#1footins\endcsname=\csuse{Xbeforenotes@#1}%
2825   \advance\skip\csname mp#1footins\endcsname by\csuse{Xafterrule@#1}%
2826 \fi
2827 }
2828 %

```

`\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.

```
2829 \providecommand{\footfudgefiddle}{64}
2830 %
```

`\para@footsetup` `\footparagraph` calls the `\para@footsetup` macro to calculate a special fudge factor, which is the ratio of the `\baselineskip` to the `\hsize`. We assume that the proper value of `\baselineskip` for the footnotes (normally 9pt) has been set already. 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`.

```
2831 \newcommand*{\para@footsetup}[1]{\csuse{Xhookgroup@#1}\csuse{
Xnotefontsize@#1}
2832 \setXnoteswidthliketwocolumns@{#1}%
2833 \ifcsemtty{Xwidth@#1}%
2834 {}%
2835 {\columnwidth=\expandafter\dimexpr\csuse{Xwidth@#1}\relax}%
2836 \dimen0=\baselineskip
2837 \multiply\dimen0 by 1024
2838 \divide \dimen0 by \columnwidth \multiply\dimen0 by \footfudgefiddle\
relax
2839 \csxdef{#1footfudgefactor}{%
2840 \expandafter\strip@pt\dimen0 }}
2841
2842 %
```

`\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.

```
2843 \newcommand*{\parafootstart}[1]{%
2844 \rightskip=0pt \leftskip=0pt%
2845 \nottoggle{Xparindent@#1}{\parindent=\z@}{}%
2846 \ifdimequal{0pt}{\Xprenotes@}{}%
2847 {%
2848 \iftoggle{Xprenotes@}{%
2849 \togglefalse{Xprenotes@}%
2850 \skip\csname #1footins\endcsname=%
2851 \glueexpr\csuse{Xprenotes@}+\csuse{Xafterrule@#1}\relax%
2852 }%
```

```

2853     {}%
2854   }%
2855   \vskip\skip\csize #1footins\endcsname%
2856   \setXnoteswidthliketwocolumns@{#1}%
2857   \setXnotespositionliketwocolumns@{#1}%
2858   \print@Xfootnoterule{#1}%
2859   \let\bid@RTL@everypar\@empty%
2860   \noindent\leavevmode}
2861 %

```

`\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>29</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>30</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>31</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>32</sup>

<sup>29</sup>Michael Downes, 'Line Breaking in \unhboxed Text', *TUGboat* 11 (1990), pp. 605–612.

<sup>30</sup>See *The TeXbook*, p. 455 (editions after January 1990).

<sup>31</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>32</sup>'Line Breaking', p. 610.

In other words, be very careful not to use `\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. 170 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.

```

2862 \newcommand*{\paravfootnote}[2]{%
2863   \csuse{Xbeforeinserting@#1}%
2864   \insert\csname #1footins\endcsname
2865   \bgroup
2866     \csuse{Xnotefontsize@#1}
2867     \footsplitskips
2868     \setbox0=\vbox{\hsize=\maxdimen%
2869       \let\bidir@RTL@everypar\@empty%
2870       \insert@Xtxtbeforenotes{#1}%
2871       \noindent\csuse{Xbhooknote@#1}%
2872       \csname #1footfmt\endcsname #2{#1}}%
2873     \setbox0=\hbox{\Xunvxh{0}{#1}}%
2874     \dp0=0pt
2875     \ht0=\csname #1footfudgefactor\endcsname\wd0
2876   %

```

Here we produce the contents of the footnote from box 0, and add a penalty of 0 between boxes in this insert.

```

2877   \if@RTL\noindent \leavevmode\fi\box0%
2878   \penalty0
2879   \egroup}
2880 %
2881 %

```

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  $\text{\TeX}$  attempts to split foot paragraphs. After trying out such a split (see *The TeXbook*, p. 124),  $\text{\TeX}$  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.

```

2882 \newcommand*\mpparavfootnote}[2]{%
2883   \global\setbox\@nameuse{mp#1footins}\vbox{%
2884     \unvbox\@nameuse{mp#1footins}%
2885     \csuse{Xnotefontsize@#1}%
2886     \footplitskips
2887     \setbox0=\vbox{\hsize=\maxdimen%
2888       \let\bidir@RTL@everypar\@empty%
2889       \insert@Xtxtbeforenotes{#1}%
2890       \noindent\color@begingroup%
2891       \csuse{Xhooknote@#1}%
2892       \csname #1footfmt\endcsname #2{#1}\color@endgroup}%
2893     \setbox0=\hbox{\Xunvxh{0}{#1}}%
2894     \dp0=\z@
2895     \ht0=\csname #1footfudgefactor\endcsname\wd0
2896     \box0
2897     \penalty0
2898   }}
2899
2900 %

```

**\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`.

```

2901 \newcommand*\Xunvxh}[2]{%
2902   \setbox0=\vbox{\unvbox#1%
2903     \global\setbox1=\lastbox}%
2904   \unhbox1
2905   \unskip           % remove \rightskip,
2906   \unskip           % remove \parfillskip,
2907   \unpenalty        % remove \penalty of 10000,
2908   \hskip\csuse{Xafternote@#2}\relax}% add the glue to go between the notes
2909
2910 %

```

**\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).

```

2911 \newcommand*\parafootfmt}[4]{%
2912   \Xstorelineinfo{#1}{#4}%
2913   \Xinsertparafootsep{#4}%
2914   \ledsetnormalparstuff@common%
2915   \printlinefootnote{#1}{#4}%

```



```

2916 \print@lemma{#1}{#2}{#4}%
2917 \csuse{Xwrapcontent@#4}{#3}%
2918 \penalty-10 }
2919 %

```

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 `\paravfootnote`.

The call to `\Xnotefontsize@<s>` is to ensure that the correct `\baselineskip` for the footnotes is used. The argument is the note series letter.

```

2920 \newcommand*{\parafootgroup}[1]{%
2921   \hsize=\expandafter\dimexpr\csuse{Xwidth@#1}\relax%
2922   \unvbox\curname #1footins\endcurname
2923   \ifcsstring{Xragged@#1}{L}{\RaggedLeft}{}%
2924   \ifcsstring{Xragged@#1}{R}{\RaggedRight}{}%
2925   \makehboxofhboxes
2926   \setbox0=\hbox{\unhbox0 \removehboxes}%
2927   \csuse{Xbhookgroup@#1}%
2928   \csuse{Xnotefontsize@#1}%
2929   \unhbox0\par%
2930   \global\hsize=\old@hsize%
2931   }%
2932
2933 %

```

**`\mpparafootgroup`** The minipage version.

```

2934 \newcommand*{\mpparafootgroup}[1]{%
2935   \setXnoteswidthliketwocolumns@{#1}%
2936   \vskip\skip\@nameuse{mp#1footins}
2937   \ifl@dpairing\ifparledgroup%
2938     \leavevmode\marks\parledgroup@{begin}%
2939     \marks\parledgroup@series{#1}%
2940     \marks\parledgroup@type{Xfootnote}%
2941   \fi\fi\normalcolor
2942   \ifparledgroup%
2943     \ifl@dpairing%
2944     \else%
2945       \setXnoteswidthliketwocolumns@{#1}%
2946       \setXnotespositionliketwocolumns@{#1}%
2947       \print@Xfootnoterule{#1}%%
2948     \fi%
2949   \else%
2950     \setXnoteswidthliketwocolumns@{#1}%

```

```

2951 \setXnotespositionliketwocolumns@{#1}%
2952 \print@Xfootnoterule{#1}%
2953 \fi%
2954 \unvbox\csname mp#1footins\endcsname
2955 \ifcsstring{Xragged@#1}{L}{\RaggedLeft}{}%
2956 \ifcsstring{Xragged@#1}{R}{\RaggedRight}{}%
2957 \makehboxofhboxes
2958 \setbox0=\hbox{\unhbox0 \removehboxes}%
2959 \csuse{Xbhookgroup@#1}%
2960 \csuse{Xnotefontsize@#1}%
2961 \nottoggle{Xparindent@#1}{\parindent=\z@}{}%
2962 \unhbox0\par}}
2963
2964 %

```

And finally, the two macros which are required to transform the long horizontal box stored in the insert' box to a printable text.

```

\makehboxofhboxes 2965 \newcommand*{\makehboxofhboxes}{\setbox0=\hbox{}}%
\removehboxes 2966 \loop
2967 \unpenalty
2968 \setbox2=\lastbox
2969 \ifhbox2
2970 \setbox0=\hbox{\box2\unhbox0}%
2971 \repeat}
2972
2973 \newcommand*{\removehboxes}{\setbox0=\lastbox
2974 \ifhbox0{\removehboxes}\unhbox0 \fi}
2975
2976 %

```

**Insertion of the footnotes separator** The command `\Xinsertparafootsep{<series>}` must be called at the beginning of `\parafootftm`.

```

\prevpage@num 2977 \newcommand{\Xinsertparafootsep}[1]{%
\Xinsertparafootsep 2978 \ifledRcol@%
2979 \ifnumequal{\csuse{#1prevpage@numR}}{\page@numR}%
2980 {\ifcsdef{prevline#1}% Be sur \prevline#1 exists.
2981 {\ifcsequal{prevline#1}{lineinfo@}%
2982 {\ifcsequal{Xsymlinenum@#1}{\csuse{Xparafootsep@#1}}{}}%
2983 {\csuse{Xparafootsep@#1}}%
2984 }%
2985 {\csuse{Xparafootsep@#1}}%
2986 }%
2987 }%
2988 \global\csname #1prevpage@numR\endcsname=\page@numR%
2989 \else%
2990 \ifnumequal{\csuse{#1prevpage@num}}{\page@num}%

```

```

2991     {\ifcsdef{prevline#1}% Be sur \prevline#1 exists.
2992       {\ifcsequal{prevline#1}{lineinfo@}%
2993         {\ifcseempty{Xsymlinenum@#1}{\csuse{Xparafootsep@#1}}{}}%
2994         {\csuse{Xparafootsep@#1}}%
2995       }%
2996     {\csuse{Xparafootsep@#1}}%
2997   }%
2998   {}%
2999   \global\csname #1prevpage@num\endcsname=\page@num%
3000   \fi%
3001 }
3002 %

```

### 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 a number (#2) of columns, each with a space (#3) between the top baseline and the top of the `\vbox`. The `\rigidbalance` macro is taken from *The TeXbook*, p. 397, with a slight change to the syntax of the arguments so that they 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`.

We do not call directly `\rigidbalance`, but we call `\Xrigidbalance` for critical notes and `\rigidbalanceX` for familiar notes. Both of them call `\rigidbalance`.

```

3003 \newcount\@k \newdimen\@h
3004 \newcommand*{\Xrigidbalance}[3]{%
3005   \hsize=\expandafter\dimexpr\csuse{Xwidth@}\@currentseries\relax%
3006   \rigidbalance{#1}{#2}{#3}%
3007 }%
3008
3009 \newcommand*{\rigidbalanceX}[3]{%
3010   \hsize=\expandafter\dimexpr\csuse{widthX@}\@currentseries\relax%
3011   \rigidbalance{#1}{#2}{#3}%
3012 }%
3013
3014 \newcommand*{\rigidbalance}[3]{%
3015   \setbox0=\box#1 \@k=#2 \@h=#3%
3016   @@line{\splittopskip=\@h \vbadness=\@M \hfilneg
3017     \valign{##\vfil\cr\dosplits}}
3018
3019 \newcommand*{\dosplits}{\ifnum\@k>0 \noalign{\hfil}\splitoff
3020   \global\advance\@k-1\cr\dosplits\fi}
3021
3022 \newcommand*{\splitoff}{\dimen0=\ht0

```

```

3023 \divide\dimen0 by\@k \advance\dimen0 by\@h
3024 \setbox2 \vsplit0 to \dimen0
3025 \unvbox2 }
3026
3027 %

```

### Three columns

```

\Xarrangement@threecol 3028 \newcommand*\Xarrangement@threecol}[1]{%
3029 \csgdef{series@display#1}{threecol}
3030 \expandafter\let\csname v#1footnote\endcsname=\threecolvfootnote
3031 \expandafter\let\csname #1footfmt\endcsname=\threecolfootfmt
3032 \expandafter\let\csname #1footgroup\endcsname=\threecolfootgroup
3033 \dimen\csname #1footins\endcsname=\csuse{Xmaxhnotes@#1}%
3034 \skip\csname #1footins\endcsname=\csuse{Xbeforenotes@#1}%
3035 \advance\skip\csname #1footins\endcsname by\csuse{Xafterrule@#1}%
3036 \threecolfootsetup{#1}
3037 %

```

The additional setup for minipages.

```

3038 \ifnoledgroup@else
3039 \expandafter\let\csname mpv#1footnote\endcsname=\mpnormalvfootnote
3040 \expandafter\let\csname mp#1footgroup\endcsname=\mpthreecolfootgroup
3041 \skip\csname mp#1footins\endcsname=\csuse{Xbeforenotes@#1}%
3042 \advance\skip\csname mp#1footins\endcsname by\csuse{Xafterrule@#1}%
3043 \mpthreecolfootsetup{#1}
3044 \fi
3045 }
3046
3047 %

```

The `\footstart` and `\footnoterule` macros for these notes assume the normal values (XII.6.2 p. 170 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.

```

3048 \newcommand*{\threecolfootsetup}[1]{%
3049   \count\csname #1footins\endcsname 333
3050   \csxdef{default@#1footins}{333}%Use this to confine the notes to one
side only
3051   \multiply\dimen\csname #1footins\endcsname \thr@@}
3052   %

```

`\mpthreecolfootsetup` The setup for minipages.

```

3053 \newcommand*{\mpthreecolfootsetup}[1]{%
3054   \count\csname mp#1footins\endcsname 333
3055   \multiply\dimen\csname mp#1footins\endcsname \thr@@}
3056   %
3057   %

```

`\threecolvfootnote` `\threecolvfootnote` This is the `\vfootnote` command for three-column notes. However, most of the code is deported on `\threecolvfootnote@inserted`. 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 #2 the full text of the note (including numbers, lemma and text).

```

3058 \notbool{parapparatus@}{\newcommand*}{\newcommand}{\threecolvfootnote}[2]{%
3059   \iftoggle{Xgroupbyline@#1}{%
3060     \prepare@Xgroupbyline{#1}{#2}{\threecolvfootnote@inserted}%
3061   }%
3062   {%
3063     \X@beforeinsertion{#1}%
3064     \insert\csname #1footins\endcsname{%
3065       \threecolvfootnote@inserted{#1}{#2}%
3066     }%
3067   }%
3068 }%
3069 %

```

```

\threecolvfootnote@inserted 3070 \notbool{parapparatus@}{\newcommand*}{\newcommand}{\threecolvfootnote@inserted}[2]{%
3071   \hsize=\expandafter\dimexpr\csuse{Xwidth@#1}\relax%
3072   \noindent\csuse{Xhooknote@#1}%
3073   \csuse{Xnotefontsize@#1}%
3074   \footssplittopskip%
3075   \csname #1footfmt\endcsname #2{#1}%
3076 }%
3077 %

```

`\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.

```

3078 \notbool{parapparatus@}{\newcommand*}{\newcommand}{\threecolfootfmt}[4]{%
3079   \threecol@begin@insert{#4}%
3080   \hspace{\parindent}%
3081   \printlinefootnote{#1}{#4}%
3082   \print@lemma{#1}{#2}{#4}%
3083   \csuse{Xwrapcontent@#4}{#3}%
3084   \nottoggle{Xgroupbyline@#4}%
3085     {\strut\par\allowbreak}%
3086   }%
3087 }%
3088 %

```

`\threecol@begin@insert` The `\threecol@begin@insert` contains code used at the beginning of any `\insert` for critical footnotes in three columns. It is used both by `\threecolfootfmt` and by `\insert@Xtxtbeforenotes`.

```

3089 \newcommand{\threecol@begin@insert}[1]{%
3090   \normal@pars%
3091   \nottoggle{Xgroupbyline@#1}%
3092     {\hspace{\csuse{Xhsizethreecol@#1}}%
3093     }%
3094   \nottoggle{Xparindent@#1}{\parindent=\z@}{%
3095     \tolerance=5000%
3096     \hangindent=\csuse{Xhangindent@#1}%
3097     \everypar{\hangindent=\csuse{Xhangindent@#1}}%
3098     \@tempdima=\parindent%
3099     \csuse{Xcolalign@#1}%
3100     \parindent=\@tempdima%
3101     \strut%
3102   }%
3103 %

```

`\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.

```

3104 \newcommand*{\threecolfootgroup}[1]{%
3105   \csuse{Xbhookgroup@#1}\par%
3106   \splittopskip=\ht\strutbox

```

```

3107 \expandafter
3108 \Xrigidbalance\csname #1footins\endcsname \thr@@ \splittopskip}
3109 %

```

**\mpthreecolfootgroup** The setup for minipages.

```

3110 \newcommand*{\mpthreecolfootgroup}[1]{%
3111 \vskip\skip\@nameuse{mp#1footins}
3112 \ifl@dpairing\ifparledgroup%
3113 \leavevmode\marks\parledgroup@{begin}%
3114 \marks\parledgroup@series{#1}%
3115 \marks\parledgroup@type{Xfootnote}%
3116 \fi\fi\normalcolor
3117 \ifparledgroup%
3118 \ifl@dpairing%
3119 \else%
3120 \setXnoteswidthliketwocolumns@{#1}%
3121 \setXnotespositionliketwocolumns@{#1}%
3122 \print@Xfootnoterule{#1}%
3123 \fi%
3124 \else%
3125 \setXnoteswidthliketwocolumns@{#1}%
3126 \setXnotespositionliketwocolumns@{#1}%
3127 \print@Xfootnoterule{#1}%
3128 \fi%
3129 \csuse{Xhookgroup@#1}\par%
3130 \splittopskip=\ht\strutbox
3131 \expandafter
3132 \Xrigidbalance\csname mp#1footins\endcsname \thr@@ \splittopskip}}
3133 %
3134 %

```

## Two columns

```

\Xarrangement@twocol 3135 \newcommand*{\Xarrangement@twocol}[1]{%
3136 \csgdef{series@display#1}{twocol}
3137 \expandafter\let\csname v#1footnote\endcsname=\twocolvfootnote
3138 \expandafter\let\csname #1footfmt\endcsname=\twocolfootfmt
3139 \expandafter\let\csname #1footgroup\endcsname=\twocolfootgroup
3140 \dimen\csname #1footins\endcsname=\csuse{Xmaxhnotes@#1}%
3141 \skip\csname #1footins\endcsname=\csuse{Xbeforenotes@#1}%
3142 \advance\skip\csname #1footins\endcsname by\csuse{Xafterrule@#1}%
3143 \twocolfootsetup{#1}
3144 %

```

The additional setup for minipages.

```

3145 \ifnoledgroup\else
3146 \expandafter\let\csname mpv#1footnote\endcsname=\mpnormalvfootnote
3147 \expandafter\let\csname mp#1footgroup\endcsname=\mptwocolfootgroup

```

```

3148 \skip\csname mp#1footins\endcsname=\csuse{Xbeforenotes@#1}%
3149 \advance\skip\csname mp#1footins\endcsname by\csuse{Xafterrule@#1}%
3150 \mptwocolfootsetup{#1}
3151 \fi
3152 }
3153
3154 %

```

`\twocolfootsetup` Here is a series of macros which are very similar to their three-column counterparts. In this case, each note is assumed to contribute only a half a line of text. And the notes are set in columns giving a gap between them of one tenth of the `\hsz`.

```

\begin{macrocode}
\newcommand*\twocolfootnote[2]{%
  \twocolfootnote@inserted
  \twocolfootfmt
  \twocolfootgroup
  \newcommand*\twocolfootsetup[1]{%
    \count\csname #1footins\endcsname 500
    \csxdef{default@#1footins}{500}%
    \multiply\dimen\csname #1footins\endcsname \tw@}
  %
  \notbool{parapparatus@}{\newcommand*{\newcommand}{\twocolfootnote}[2]{%
    \iftoggle{Xgroupbyline@#1}{%
      \prepare@Xgroupbyline{#1}{#2}{\twocolfootnote@inserted}%
    }{%
      \X@beforeinsertion{#1}%
      \insert\csname #1footins\endcsname{%
        \twocolfootnote@inserted{#1}{#2}%
      }%
    }%
  }}%
  \notbool{parapparatus@}{\newcommand*{\newcommand}{\twocolfootnote@inserted}[2]{%
    \hsz=\expandafter\dimexpr\csuse{Xwidth@#1}\relax%
    \noindent\csuse{Xhooknote@#1}%
    \csuse{Xnotefontsize@#1}%
    \footsplitskips%
    \csname #1footfmt\endcsname #2{#1}%
  }}%
  %
  \notbool{parapparatus@}{\newcommand*{\newcommand}{\twocolfootfmt}[4]{% 4th
    arg is optional, for backward compatibility
    \twocol@begin@insert{#4}%
    \hspace{\parindent}%
    \printlinefootnote{#1}{#4}%
    \print@lemma{#1}{#2}{#4}%
    \csuse{Xwrapcontent@#4}{#3}%
    \nottoggle{Xgroupbyline@#4}{%
      {\strut\par\allowbreak}%
    }%
  }}%
  %
\end{macrocode}

```



```

3189 \newcommand{\twocol@begin@insert}[1]{%
3190   \normal@pars%
3191   \hsize \csuse{Xhsizetwocol@#1}%
3192   \nottoggle{Xparindent@#1}{\parindent=\z@}{}%
3193   \tolerance=5000%
3194   \hangindent=\csuse{Xhangindent@#1}%
3195   \everypar{\hangindent=\csuse{Xhangindent@#1}}%
3196   \@tempdima=\parindent%
3197   \csuse{Xcolalign@#1}%
3198   \parindent=\@tempdima%
3199   \strut%
3200 }%
3201
3202 \newcommand*{\twocolfootgroup}[1]{%
3203   \csuse{Xbhookgroup@#1}\par%
3204   \splittopskip=\ht\strutbox
3205   \expandafter
3206   \Xrigidbalance\csname #1footins\endcsname \tw@ \splittopskip}
3207
3208 %

```

`\mptwocolfootsetup` The versions for minipages.

`\mptwocolfootgroup`

```

3209 \newcommand*{\mptwocolfootsetup}[1]{%
3210   \count\csname mp#1footins\endcsname 500
3211   \multiply\dimen\csname mp#1footins\endcsname \tw@}
3212 %

```

```

3213 \newcommand*{\mptwocolfootgroup}[1]{%
3214   \vskip\skip\@nameuse{mp#1footins}
3215   \ifl@dpairing\ifparledgroup%
3216     \leavevmode\marks\parledgroup@{begin}%
3217     \marks\parledgroup@series{#1}%
3218     \marks\parledgroup@type{Xfootnote}%
3219   \fi\fi\normalcolor
3220   \ifparledgroup%
3221     \ifl@dpairing%
3222     \else%
3223       \setXnoteswidthliketwocolumns@{#1}%
3224       \setXnotespositionliketwocolumns@{#1}%
3225       \print@Xfootnoterule{#1}%
3226     \fi%
3227   \else%
3228     \setXnoteswidthliketwocolumns@{#1}%
3229     \setXnotespositionliketwocolumns@{#1}%
3230     \print@Xfootnoterule{#1}%
3231   \fi%
3232   \csuse{Xbhookgroup@#1}\par%
3233   \splittopskip=\ht\strutbox
3234   \expandafter

```

```

3235 \Xrigidbalance\csname mp#1footins\endcsname \tw@ \splittopskip}}
3236
3237 %

```

## 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.

Note that these commands are not directly called by `reledmac`, but are enclosed as default value of specific hooks. Consequently, people should not redefine them, but use instead the `\Xlinrangeseparator`, `\Xendlinrangeseparator`, `\Xsublinesep`, `\Xendsublinesep` and `\Xlemmaseparator` macros.

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.

```

3238 \def\endashchar{\textnormal{--}}
3239
3240 \newcommand*\fullstop{\textnormal{.}}
3241 \def\Xsublinesep@side{\fullstop}
3242
3243 \newcommand*\rbracket{\textnormal{%
3244   \csuse{text\csuse{footnote@lang}}{%
3245     \ifluatex%
3246       \ifdefstring{\footnote@luatextextdir}{TRT}{\thinspace[]}{\thinspace
3247     }}%
3248     \else%
3249     \thinspace}%
3250     \fi}%
3251   }}
3252
3253 %

```

### XII.7.2 Pstart number in footnote

`\printpstart` The `\printpstart` macro prints the pstart number for a note.

```

3254 \newcommand{\printpstart}[0]{%
3255   \ifboolexpr{bool{l@dpairing} or bool{l@dprintingpages} or bool{
l@dprintingcolumns}}{%
3256     \ifledRcol%
3257       \thepstartR%
3258     \else%
3259       \thepstartL%
3260     \fi%
3261   }{%
3262     \thepstart%
3263   }%
3264 }
3265 %

```

### XII.7.3 Lemma printing

`\print@lemma` `\print@lemma` is called inside critical footnotes to print the lemma and the lemma separator (#1: line number and font information, #2: lemma, #3: series).

```

3266 %
3267 \newcommand{\print@lemma}[3]{%
3268   \bgroup%
3269   \nottoggle{Xlemmadisablefontselection@#3}%
3270   {\select@lemmafont#1|}%
3271   }%
3272   \bgroup%
3273   \csuse{Xlemmafont@#3}%Deprecated
3274   \csuse{Xwraplemma@#3}{#2}%
3275   \egroup%
3276   \egroup%
3277   \iftoggle{nosep@}{%
3278     \hskip\csuse{Xinplaceoflemmaseparator@#3}%
3279     \relax%
3280   }%
3281   {\ifcsemt{Xlemmaseparator@#3}%
3282     {%
3283       \hskip\csuse{Xinplaceoflemmaseparator@#3}%
3284       \relax%
3285     }%
3286     {%
3287       \nobreak%
3288       \hskip\csuse{Xbeforelemmaseparator@#3}%
3289       \csuse{Xlemmaseparator@#3}%
3290       \hskip\csuse{Xafterlemmaseparator@#3}%
3291       \relax%
3292     }%

```

```

3293 }%
3294 }%
3295 %

```

### XII.7.4 Line number printing

**\Xstorelineinfo** The `\Xstorelineinfo` macro is used to store some data about line number of the current critical footnote, data which will be reused later for the `\Xnumberonlyfirstinline` and related setting.

#1 footnote specification for the current footnote ; #2 footnote series.

```

3296 \newcommand{\Xstorelineinfo}[2]{%
3297   \l@dp@rsefootspec#1|{%
3298     \iftoggle{Xnumberonlyfirstintwolines@#2}{%
3299       \xdef\lineinfo@{\l@dparsedstartline - \l@dparsedstartsub - \
3300       \l@dparsedendline - \l@dparsedendsub}%
3301     }%
3302     {%
3303       \xdef\lineinfo@{\l@dparsedstartline - \l@dparsedstartsub}%
3304     }%
3305   }%
3306   %

```

**\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 `\printlinefootnotenumbers`.

```

3306 \newcommand{\printlinefootnote}[2]{%
3307   \iftoggle{nonum@}{%Try if the line number must printed for this specific
3308   not (by default, yes)
3309   \hspace{\csuse{Xinplaceofnumber@#2}}%
3310   }%
3311   {%
3312     \iftoggle{Xnonumber@#2}{%Try if the line number must printed (by
3313     default, yes)
3314     {%
3315       \hspace{\csuse{Xinplaceofnumber@#2}}%
3316     }%
3317     {%
3318       \iftoggle{Xnumberonlyfirstinline@#2}{% If for this series the
3319       line number must be printed only in the first time.
3320       {%
3321         \ifcsdef{prevline#2}%
3322         {%Be sure the \prevline exists.
3323         \ifcsequal{prevline#2}{\lineinfo@}{%Try it
3324         {%
3325         \ifcsequal{Xsymlinenum@#2}%

```

```

3324         {%
3325         \hspace{\csuse{Xinplaceofnumber@#2}}}%
3326         }%
3327         {\printsymlinefootnotearea{#2}}}%
3328         }%
3329         {%
3330         \printlinefootnotearea{#1}{#2}%
3331         }%
3332         }%
3333         {%
3334         \printlinefootnotearea{#1}{#2}%
3335         }%
3336         }%
3337         {%
3338         \printlinefootnotearea{#1}{#2}%
3339         }%
3340         \csxdef{prevline#2}{\lineinfo@}%
3341         }%
3342     }%
3343 }%
3344 }%
3345 }
3346 %

```

**\printsymlinefootnotearea** This macro prints the space before the line symbol, changes the font, when prints the line symbol and the space after it.

```

3347 \newcommand{\printsymlinefootnotearea}[1]{%
3348 \hspace{\csuse{Xbeforesymlinenum@#1}}}%
3349 \csuse{Xnotenumfont@#1}%
3350 \ifdimequal{\csuse{Xboxsymlinenum@#1}}{\z@}%
3351 {\csuse{Xsymlinenum@#1}}}%
3352 {\hbox to \csuse{Xboxsymlinenum@#1}%
3353 {\csuse{Xsymlinenum@#1}\hfill}}%
3354 }%
3355 \hspace{\csuse{Xaftersymlinenum@#1}}}%
3356 }%
3357 %

```

**\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.)

```

3358 \newcommand{\printlinefootnotearea}[2]{%
3359 \printXbeforenumber{#2}%
3360 \csuse{Xnotenumfont@#2}%
3361 \boxfootnotenumbers{#1}{#2}%
3362 \printXafternumber{#2}%
3363 }%

```

```
3364 %
```

**\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.

```
3365 \newcommand{\boxfootnotenumbers}[2]{%
3366   \ifdimequal{\csuse{Xboxlinenum@#2}}{0pt}{%
3367     \printlinefootnotenumbers{#1}{#2}%
3368   }%
3369   {%
3370     \hbox to \csuse{Xboxlinenum@#2}%
3371     {%
3372       \IfSubStr{RC}{\csuse{Xboxlinenumalign@#2}}{\hfill}{%
3373         \printlinefootnotenumbers{#1}{#2}%
3374       \IfSubStr{LC}{\csuse{Xboxlinenumalign@#2}}{\hfill}{%
3375         }%
3376     }%
3377   }%
3378   %
```

**\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.

```
3379 \newcommand{\printlinefootnotenumbers}[2]{%
3380   \xdef\@currentseries{#2}%
3381   \ifboolexpr{%
3382     (togl{Xpstart@#2} and bool{numberpstart})%
3383     or togl{Xpstarteverytime@#2}}%
3384   {\printpstart}{%
3385     \iftoggle{Xstanza@#2}{%
3386       \ifnumberstanza%
3387         \printstanza%
3388         \csuse{Xstanzaseparator@#2}%
3389       \fi%
3390     }{%
3391       \iftoggle{Xonlypstart@#2}{%
3392         \csuse{Xtxtbeforenumber@#2}%
3393         \printlines#1|\ifledRcol@\@Rlineflag\fi}%
3394     }%
3395   %
```

**\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.)

```
3396 \newcommand{\printXbeforenumber}[1]{%
3397   \hspace{\csuse{Xbeforenumber@#1}}%
3398 }%
3399 %
```

`\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

```

3400 \newcommand{\printXafternumber}[1]{%
3401   \iftoggle{Xnonbreakableafternumber@#1}{\nobreak}{}%
3402   \hspace{\csuse{Xafternumber@#1}}%
3403 }%
3404 %

```

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. 99: 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@Xtwolines` and `\ifl@d@Xmorethantwolines` to print a symbol which stands for “and subsequent” when there are two, three or more lines.

```

\ifl@d@pnum05 \newif\ifl@d@pnum
\ifl@d@ssub06 \newif\ifl@d@ssub
\ifl@d@elin07 \newif\ifl@d@elin
\ifl@d@esl08 \newif\ifl@d@esl
\ifl@d@dash09 \newif\ifl@d@dash
\ifl@d@Xtwolines3410 \newif\ifl@d@Xtwolines%
\ifl@d@Xmorethantwolines3411 \newif\ifl@d@Xmorethantwolines%
3412 %

```

`\l@dparsefootspec` `\l@dparsefootspec{<spec>}{<lemma>}{<text>}` parses a footnote specification. `<lemma>` and `<text>` are the lemma and text respectively. `<spec>` is the line and page number and lemma font specifier in `\l@d@nums` style format. The real work is done by `\l@dparsefootspec` which defines macros holding the numeric values. In many cases, this last command is called directly. Just a reminder of the arguments:

<code>\l@dparsefootspec</code>	<code>#1</code>	<code> </code>	<code>#2</code>	<code> </code>	<code>#3</code>	<code> </code>	<code>#4</code>	<code> </code>	<code>#5</code>	<code> </code>	<code>#6</code>	<code> </code>	<code>#7</code>
<code>\l@dparsefootspec</code>	start-page		line		subline		end-page		line		subline		fontflag

```

3413 \newcommand*{\l@dp@rsefootsec}[3]{\l@dp@rsefootsec#1|}
3414 \def\l@dp@rsefootsec#1|#2|#3|#4|#5|#6|#7|{%
3415   \gdef\l@dparsedstartpage{#1}%
3416   \gdef\l@dparsedstartline{#2}%
3417   \gdef\l@dparsedstartsub{#3}%
3418   \gdef\l@dparsedendpage{#4}%
3419   \gdef\l@dparsedendline{#5}%
3420   \gdef\l@dparsedendsub{#6}%
3421 }
3422 %

```

Initialise the several number value macros.

```

3423 \def\l@dparsedstartpage{0}%
3424 \def\l@dparsedstartline{0}%
3425 \def\l@dparsedstartsub{0}%
3426 \def\l@dparsedendpage{0}%
3427 \def\l@dparsedendline{0}%
3428 \def\l@dparsedendsub{0}%
3429
3430 %

```

**\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`.

```

3431 \newcommand*{\setprintlines}[6]{%
3432   \l@d@pnumfalse \l@d@dashfalse
3433 %

```

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

```

3434   \ifbypage@
3435     \ifnum#4=#1 \else
3436       \l@d@pnumtrue
3437       \l@d@dashtrue
3438     \fi
3439   \fi
3440 %

```

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.

```

3441   \ifl@d@pnum \l@d@elintrue \else \l@d@elinfalse \fi
3442   \ifnum#2=#5 \else
3443     \l@d@elintrue
3444     \l@d@dashtrue
3445   \fi
3446 %

```

We print the starting sub-line if it is nonzero.



```

3447 \l@d@ssubfalse
3448 \ifnum#3=0 \else
3449     \l@d@ssubtrue
3450 \fi
3451 %

```

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.

```

3452 \l@d@eslfalse
3453 \ifnum#6=0 \else
3454     \ifnum#6=#3
3455         \ifl@d@elin \l@d@esltrue \else \l@d@eslfalse \fi
3456     \else
3457         \l@d@esltrue
3458         \l@d@dashtrue
3459     \fi
3460 \fi%
3461 %

```

However, if the \Xtwolines is set for the current series, we do not print the last line number.

```

3462 \ifl@d@dash%
3463     \ifboolexpr{togl{fulllines@} or test{\ifcsemt{Xtwolines@}\
3464     @currentseries}}}%
3465     {%
3466     \setistwofollowinglines{#1}{#2}{#4}{#5}%
3467     \ifboolexpr{%
3468         (%
3469             togl {Xtwolinesbutnotmore@\@currentseries}%
3470             and not%
3471             (%
3472                 bool {istwofollowinglines@}%
3473             )%
3474         )%
3475     or%
3476     (%
3477         (not test{\ifnumequal{#1}{#4}})%
3478         and togl{Xtwolinesonlyinsamepage@\@currentseries}%
3479     )%
3480     }%
3481     {%
3482     \l@d@dashfalse%
3483     \l@d@Xtwolinestruetrue%
3484     \l@d@elinfalse%
3485     \l@d@eslfalse%
3486     \ifcsemt{Xmorethantwolines@\@currentseries}%
3487     {%
3488

```

```

3489         {\ifistwofollowinglines@else%
3490          \l@d@Xmorethantwolines>true%
3491          \fi%
3492        }%
3493      }%
3494    }%
3495  \fi%
3496 %

```

End of \setprintlines.

```

3497 }%
3498 %

```

**\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.

```

3499 \newif\ifistwofollowinglines@%
3500 \newcommand{\setistwofollowinglines}[4]{%
3501   \ifcsdef{lastlinenumberon@#1}%
3502     {\numdef{\tmp}{\csuse{lastlinenumberon@#1}}}%
3503     {\numdef{\tmp}{0}}}%
3504   \istwofollowinglines@false%
3505   \ifnumequal{#4-#2}{1}%
3506     {\istwofollowinglines@true}%
3507   {\ifbypage@%
3508     \ifnumequal{#3-#1}{1}%
3509     {%
3510       \ifnumequal{#2}{\tmp}%
3511       {\ifnumequal{#4}{1}{\istwofollowinglines@true}}}%
3512     }%
3513   }%

```

```

3514     {}%
3515     \fi%
3516   }%
3517 }%
3518 %

```

`\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. Arguments are 1) start page number 2) start line number 3) start subline number 4) end page number 5) end line number 6) end subline number 7) font specification 8) side flag

```

3519 \def\printlines#1|#2|#3|#4|#5|#6|#7|#8|{%
3520   \begingroup%
3521   %

```

If we use Lua<sub>T</sub><sub>E</sub>X, ensure we use good text's direction.

```

3522   \ifluatex%
3523     \edef\@tmp{\the\textdir}%
3524     \ifdefstring{\@tmp}{TLT}{\textdir TLT}%Test in order to prevent
    spurious space (bug #397)
3525   \fi%
3526   %

```

Decide which part of line number components we will print.

```

3527   \setprintlines{#1}{#2}{#3}{#4}{#5}{#6}%
3528   %

```

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.

```

3529   \ifdimequal{\csuse{Xboxstartlinenum@\@currentseries}}{0pt}%
3530     {\bgroup}%
3531     {\leavevmode\hbox to \csuse{Xboxstartlinenum@\@currentseries}\bgroup\
    hfill}%
3532   \ifl@d@pnum%
3533     \wrap@edcrossref{\@this@crossref@start}{#1}%
3534     \csuse{Xpagelinesep@\@currentseries}%
3535   \fi%
3536   \wrap@edcrossref{\@this@crossref@start}{%
    \linenumrep{#2}%
3537     \iftoggle{Xlineflag@\@currentseries}{#8}{}%
3538   }%
3539   \ifl@d@ssub%
3540     \csuse{Xsublinesep@\@currentseries}%
3541     \wrap@edcrossref{\@this@crossref@start}{\sublinenumrep{#3}}%
3542   \fi
3543   \egroup%
3544   %
3545   %

```

Then print the dash + end line number, or the range symbol.

```

3546 \ifdimequal{\csuse{Xboxendlinenum@\@currentseries}}{0pt}%
3547 {\bgroup}%
3548 {\hbox to \csuse{Xboxendlinenum@\@currentseries}\bgroup}%
3549 \ifl@d@Xtwolines%
3550 \ifl@d@Xmorethantwolines%
3551 \csuse{Xmorethantwolines@\@currentseries}%
3552 \else%
3553 \csuse{Xtwolines@\@currentseries}%
3554 \fi%
3555 \else%
3556 \ifl@d@dash%
3557 \ifdefined\linrangesep%
3558 \linrangesep%
3559 \else%
3560 \csuse{Xlinrangeseparator@\@currentseries}%
3561 \fi%
3562 \fi%
3563 \ifl@d@pnum%
3564 \wrap@edcrossref{\@this@crossref@end}{#4}%
3565 \csuse{Xpagelinesep@\@currentseries}%
3566 \fi%
3567 \ifl@d@elin%
3568 \wrap@edcrossref{\@this@crossref@end}{%
3569 \linenumrep{#5}%
3570 \iftoggle{Xlineflag@\@currentseries}{#8}{}}%
3571 }%
3572 \fi%
3573 \ifl@d@esl%
3574 \ifl@d@elin%
3575 \csuse{Xsublinesep@\@currentseries}%
3576 \fi%
3577 \wrap@edcrossref{\@this@crossref@end}{\sublinenumrep{#6}}%
3578 \fi%
3579 \fi%
3580 \ifdimequal{\csuse{Xboxendlinenum@\@currentseries}}{0pt}%
3581 {}%
3582 {\hfill}%Prevent underfull hbox
3583 \egroup%
3584 \endgroup%
3585 }%
3586 %

```

### XII.7.5 Footnote grouped by line

`\prepare@Xgroupbyline` `\prepare@Xgroupbyline` is a macro called on on the `\langle XXX \rangle footnote` if `\Xgroupbyline` is set to true, instead of calling directly the `\insert`.

```

3587 \newcommand{\prepare@Xgroupbyline}[3]{%

```

```

3588 \iftoggle{Xgroupbylineseparetwolines@#1}{%
3589 \l@dparsfootspec#2%
3590 \ifcsdef{#1@forinserting@l@dparsedendpage-l@dparsedendline-l
l@dparsedendsub}%
3591 {%
3592 \csgappto%
3593 {#1@forinserting@l@dparsedendpage-l@dparsedendline-l
l@dparsedendsub}%
3594 {%
3595 \ifcsemt{Xsymlinenum@#1}%
3596 {\csuse{Xparafootsep@#1}}%
3597 {}%
3598 #3{#1}{#2}%
3599 \hskip\csuse{Xafternote@#1}\relax%
3600 }%
3601 }%
3602 {%
3603 \csdef%
3604 {#1@forinserting@l@dparsedendpage-l@dparsedendline-l
l@dparsedendsub}%
3605 {%
3606 #3{#1}{#2}%
3607 \hskip\csuse{Xafternote@#1}\relax%
3608 }%
3609 }%
3610 \listcsxadd{#1@forinserting}{l@dparsedendpage-l@dparsedendline-l
l@dparsedendsub}%
3611 }{%
3612 \ifcsdef{#1@forinserting@all}{%
3613 \csgappto%
3614 {#1@forinserting@all}%
3615 {%
3616 \ifcsemt{Xsymlinenum@#1}%
3617 {\csuse{Xparafootsep@#1}}%
3618 {}%
3619 #3{#1}{#2}%
3620 \hskip\csuse{Xafternote@#1}\relax%
3621 }%
3622 }%
3623 {%
3624 \csdef%
3625 {#1@forinserting@all}%
3626 {%
3627 #3{#1}{#2}%
3628 \hskip\csuse{Xafternote@#1}\relax%
3629 }%
3630 }%
3631 \listcsgadd{#1@forinserting}{all}%
3632 }%
3633 }%

```

```
3634 %
```

## XIII Familiar footnotes

### XIII.1 Adjacent footnotes

The original edmac provided users with five series of critical footnotes (`\Afootnote` `\Bfootnote` `\Cfootnote` `\Dfootnote` `\Efootnote`), and  $\text{\LaTeX}$  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`.

Maïeul Rouquette has maintained this feature in `reledmac`, despite he thought that is not directly in relationship with the aim of `reledmac`.

`\multiplefootnotemarker` `\multfootsep` 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`.

```
3635 \providecommand*\multiplefootnotemarker}{3sp}
3636 \providecommand*\multfootsep}{\textsuperscript{\normalfont,}}
3637
3638 %
```

`\m@mmf@prepare` A pair of self-cancelling kerns. This may have been defined in the `memoir` class.

```
3639 \providecommand*\m@mmf@prepare}{%
3640   \kern-\multiplefootnotemarker
3641   \kern\multiplefootnotemarker\relax}
3642 %
```

`\m@mmf@check` This may have been defined in the `memoir` class. If it recognises the last kern as `\multiplefootnotemarker` it typesets `\multfootsep`.

```
3643 \providecommand*\m@mmf@check}{%
3644   \ifdim\lastkern=\multiplefootnotemarker\relax
3645     \edef\@x@sf{\the\spacefactor}%
3646     \unkern
3647     \multfootsep
3648     \spacefactor\@x@sf\relax
3649   \fi}
3650
3651 %
```

We have to modify `\@footnotetext` and `\@footnotemark`. However, if `memoir` is used the modifications have already been made.

```

3652 \@ifclassloaded{memoir}{\%
3653 \%

```

**\@footnotetext** Add \m@mmf@prepare at the end of \@footnotetext.

```

3654 \apptocmd{\@footnotetext}{\m@mmf@prepare}{\%}{\%
3655 \%

```

**\@footnotemark** Modify \@footnotemark to cater for adjacent \footnotes.

```

3656 \patchcmd{\@footnotemark}
3657 {\nobreak}
3658 {\m@mmf@check
3659 \nobreak
3660 }
3661 {\%}{\%
3662 \patchcmd{\@footnotemark}
3663 {\@makefnmark}
3664 {\@makefnmark
3665 \m@mmf@prepare
3666 }
3667 {\%}{\%
3668 \%
3669 \%

```

Finished the modifications for the non-memoir case.

```

3670 }
3671
3672 \%

```

## 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 with its \@footnotetext, using different forms for when in numbered or regular text.

```

3673 \pretocmd{\@footnotetext}{\%
3674 \ifnumberedpar@
3675 \edtext{\l@dbfnote{#1}}{\%
3676 \else
3677 }{\%}{\%
3678 \apptocmd{\@footnotetext}{\fi}{\%}{\%
3679 \%

```

**\l@dbfnote** \l@dbfnote adds the footnote to the insert list, and \v1@dbfnote calls the original

**\v1@dbfnote** \@footnotetext. We also patch \footnote in order to get the correct footnote

**\v1@dbfnote** numbers when typesetting parallel texts. This is moved into a \get@fnmark command.

**\footnote**

**\get@fnmark**

**\get@thisfootnote**

```

3680
3681 \patchcmd%
3682 {\footnote}%
3683 {\stepcounter\@mpfn}%
3684 {%
3685 \ifboolexpr{bool{l@dpairing} or bool{l@dprintingpages} or bool{
3686 l@dprintingcolumns}}{%
3687 \global\advance\footnote@reading by \@ne%
3688 \get@thisfootnote%
3689 \get@fnmark{\thisc@footnote}%
3690 \ifcsdef{footnotereading\the\footnote@reading=typeset}%
3691 {\setcounter{\@mpfn}{\csuse{footnotereading\the\footnote@reading=
3692 typeset}}}%
3693 {\setcounter{\@mpfn}{\footnote@reading}}%
3694 }%
3695 \stepcounter\@mpfn%
3696 }%
3697 }%
3698
3699 \newcommand{\get@thisfootnote}{%
3700 \ifboolexpr{bool{l@dpairing} or bool{l@dprintingpages} or bool{
3701 l@dprintingcolumns}}{%
3702 \protected@xdef\thisc@footnote{\the\footnote@reading}%
3703 }%
3704 \protected@xdef\thisc@footnote{\the\c@footnote}%
3705 }%
3706
3707 \newcommand{\l@dbfnote}[1]{%
3708 \get@thisfootnote%
3709 \gdef\@tag{#1\relax}%
3710 \ifledRcol%
3711 \xright@appenditem{%
3712 \ifdefined\Hy@footnote@currentHref%
3713 \noexpand\def\noexpand\Hy@footnote@currentHref{\
3714 Hy@footnote@currentHref}%
3715 \fi%
3716 \noexpand\vl@dbfnote{{\expandonce\@tag}}{\thisc@footnote}%
3717 }%
3718 \to\inserts@listR
3719 \global\advance\insert@countR \@ne%
3720 \else%
3721 \xright@appenditem{%
3722 \ifdefined\Hy@footnote@currentHref%
3723 \noexpand\def\noexpand\Hy@footnote@currentHref{\
3724 Hy@footnote@currentHref}%
3725 \fi%
3726 \noexpand\vl@dbfnote{{\expandonce\@tag}}{\thisc@footnote}%

```



```

3725 }%
3726         \to\inserts@list
3727         \global\advance\insert@count \@ne%
3728     \fi
3729     \ignorespaces%
3730 }%
3731
3732 \newcommand{\get@fnmark}[1]{%
3733     \ifbool{bool{!@dpairing}} or bool{!@dprintingpages} or bool{
3734         !@dprintingcolumns}}{%
3735         \stepcounter{footnote@typeset}%
3736         \setcounter{footnote}{\c@footnote@typeset}%
3737         \immediate\write\@mainaux{%
3738             \csgdef{footnotereading#1=typeset}{\the\c@footnote@typeset}%
3739         }%
3740         \def\@thefnmark{\thefootnote}%
3741     }%
3742     {%
3743         \setcounter{footnote}{#1}%
3744         \def\@thefnmark{\thefootnote}%
3745     }%
3746 }%
3747
3748 \newcommand{\v1@dbfnote}[2]{%
3749     \get@fnmark{#2}%
3750     \@footnotetext{#1}%
3751 }%
3752 %

```

### 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`

```

3753 \newcommand*\prebodyfootmark{%
3754     \leavevmode
3755     \ifhmode
3756         \edef\x@sf{\the\spacefactor}%
3757         \m@mmf@check
3758         \nobreak
3759     \fi}
3760 \newcommand*\postbodyfootmark{%
3761     \m@mmf@prepare
3762     \ifhmode\spacefactor\x@sf\fi\relax}
3763
3764 %

```

## XIII.4 Footnote arrangement

### XIII.4.1 User level macro

**\arrangementX** `\arrangementX[ $\langle s \rangle$ ]{ $\langle arrangement \rangle$ }` command calls, for each series, a specific command which set many counters and commands in order to define specific arrangement.

```

3765 \newcommandx{\arrangementX}[2][1,usedefault]{%
3766   \def\do##1{%
3767     \csname arrangementX@#2\endcsname{##1}%
3768   }%
3769   \ifstrempy{#1}%
3770     {%
3771       \dolistloop{\@series}%
3772     }%
3773     {
3774       \docsvlist{#1}%
3775     }%
3776   }%
3777   %

```

### XIII.4.2 Normal footnotes

**\normal@footnotemarkX** `\normal@footnotemarkX{ $\langle series \rangle$ }` sets up the typesetting of the marker at the point where the footnote is called for.

```

3778 \newcommand*{\normal@footnotemarkX}[1]{%
3779   \prebodyfootmark
3780   \wrapped@bodyfootmarkX{#1}%
3781   \postbodyfootmark}
3782
3783 %

```

**\normalbodyfootmarkX** The `\normalbodyfootmarkX{ $\langle series \rangle$ }` *really* typesets the in-text marker. The style is the normal superscript.

```

3784 \newcommand*{\normalbodyfootmarkX}[1]{%
3785   \hbox{\textsuperscript{\normalfont\@nameuse{@thefnmark#1}}}}
3786 %

```

**\normalvfootnoteX** `\normalvfootnoteX{ $\langle series \rangle$ }{ $\langle text \rangle$ }` does the `\insert` for the  $\langle series \rangle$  and calls the series' `\footfmt...` to format the  $\langle text \rangle$ .

```

3787 \notbool{parapparatus@}{\newcommand*}{\newcommand}{\normalvfootnoteX}[2]{%
3788   \csuse{beforeinsertingX@#1}%
3789   \insert\@nameuse{footins#1}\bgroup
3790     \reset@font%
3791     \hsize=\expandafter\dimexpr\csuse{widthX@#1}\relax%
3792     \noindent\csuse{bhooknoteX@#1}%
3793     \csuse{notefontsizeX@#1}%
3794     \footssplit skips

```

```

3795 \ifl@dpairing\ifl@dpaging\else%
3796 \setnoteswidthliketwocolumnsX@{#1}%
3797 \fi\fi%
3798 \setnotesXpositionliketwocolumns@{#1}%
3799 \spaceskip=\z@skip \xspaceskip=\z@skip
3800 \csuse{\csuse{footnote@dir}}\@nameuse{footfmt#1}{#1}{#2}\egroup}
3801
3802 %

```

`\mpnormalvfootnoteX` The minipage version.

```

3803 \newcommand*{\mpnormalvfootnoteX}[2]{%
3804 \get@thisfootnoteX{#1}%
3805 \get@fnmarkX{#1}{\thisc@footnote}%
3806 \edef\thisc@footnoteX@reading{\the\csname footnote#1@reading\endcsname}%
3807 \global\setbox\@nameuse{mpfootins#1}\vbox{%
3808 \unvbox\@nameuse{mpfootins#1}
3809 \noindent\csuse{bhooknoteX@#1}%
3810 \csuse{notefontsizeX@#1}%
3811 \hsize\columnwidth
3812 \@parboxrestore
3813 \color@begingroup
3814 \@nameuse{footfmt#1}{#1}{#2}\color@endgroup}}
3815
3816 %

```

`\normalfootfmtX` `\normalfootfmtX{<series>}{<text>}` typesets the footnote text, prepended by the marker.

```

3817 \notbool{parapparatus0}{\newcommand*{\newcommand}{\normalfootfmtX}[2]{%
3818 \ifluatex%
3819 \texdir\footnote@luatextextdir%
3820 \pardir\footnote@luatexpardir%
3821 \par%
3822 \fi%
3823 \protected@edef\@currentlabel{%
3824 \@nameuse{@thefnmark#1}%
3825 }%
3826 \ledsetnormalparstuffX{#1}%
3827 \hangindent=\csuse{hangindentX@#1}%
3828 \everypar{\hangindent=\csuse{hangindentX@#1}}%
3829 \rule\z@{\splittopskip%
3830 {{\csuse{notennumfontX@#1}}\wrapped@footfootmarkX{#1}}%
3831 \csuse{wrapcontentX@#1}{#2}%
3832 \strut\par}}
3833
3834 %

```

`\normalfootfootmarkX` `\normalfootfootmarkX{<series>}` is called by `\normalfootfmtX` to typeset the footnote marker in the footer before the footnote text.

```

3835 \newcommand*{\normalfootfootmarkX}[1]{%
3836   \textsuperscript{\@nameuse{@thefnmark#1}}}%
3837
3838 %

```

**\normalfootstartX** `\normalfootstartX{<series>}` is the `<series>` footnote starting macro used in the output routine.

```

3839 \newcommand*{\normalfootstartX}[1]{%
3840   \ifdimequal{Opt}{\prenotesX@}{}%
3841   {%
3842     \iftoggle{prenotesX@}{%
3843       \togglefalse{prenotesX@}%
3844       \skip\csname footins#1\endcsname=%
3845       \glueexpr\csuse{prenotesX@}+\csuse{afterruleX@#1}\relax%
3846     }%
3847   }%
3848   }%
3849   \vskip\skip\csname footins#1\endcsname%
3850   \leftskip=\z@
3851   \rightskip=\z@
3852   \ifl@dpairing\else%
3853     \hsize=\old@hsize%
3854     \fi%
3855   \setnoteswidthliketwocolumnsX@{#1}%
3856   \setnotesXpositionliketwocolumns@{#1}%
3857   \print@footnoteXrule{#1}%
3858 }%
3859
3860 %

```

**\normalfootnoteruleX** The rule drawn before the footnote series group.

```

3861 \let\normalfootnoteruleX=\footnoterule
3862
3863 %

```

**\normalfootgroupX** `\normalfootgroupX{<series>}` sends the contents of the `<series>` insert box to the output page without alteration.

```

3864 \newcommand*{\normalfootgroupX}[1]{%
3865   \csuse{bhookgroupX@#1}%
3866   \unvbox\@nameuse{footins#1}%
3867   \hsize=\old@hsize%
3868 }%
3869
3870 %

```

**\mpnormalfootgroupX** The minipage version.

```

3871 \newcommand*{\mpnormalfootgroupX}[1]{%
3872   \vskip\skip\@nameuse{mpfootins#1}
3873   \ifl@dpairing\ifparledgroup%
3874     \leavevmode\marks\parledgroup@{begin}%
3875     \marks\parledgroup@series{#1}%
3876     \marks\parledgroup@type{footnoteX}%
3877   \fi\fi\normalcolor
3878   \ifparledgroup%
3879     \ifl@dpairing%
3880     \else%
3881       \setnoteswidthliketwocolumnsX@{#1}%
3882       \setnotesXpositionliketwocolumns@{#1}%
3883       \print@footnoteXrule{#1}%
3884     \fi%
3885   \else%
3886     \setnoteswidthliketwocolumnsX@{#1}%
3887     \setnotesXpositionliketwocolumns@{#1}%
3888     \print@footnoteXrule{#1}%
3889   \fi%
3890   \csuse{bhookgroupX@#1}%
3891   \unvbox\@nameuse{mpfootins#1}}
3892
3893 %

```

**\normalbfnoteX**<sup>94</sup>

```

3895 \newcommand{\normalbfnoteX}[2]{%
3896   \get@thisfootnoteX{#1}%
3897   \ifledRcol%
3898     \ifluatex
3899       \footnotelang@lua[R]%
3900     \fi
3901     \@ifundefined{xpg@main@language}%if polyglossia
3902     {}%
3903     {\footnotelang@poly[R]}%
3904     \xright@appenditem{%
3905       \noexpand\led@set@index@fornote{#1}%
3906       \noexpand\prepare@edindex@fornote{\led@nums}%
3907       \unexpanded{\def\this@footnoteX@reading}{\the\csname footnote#1
@reading\endcsname}%
3908       \noexpand\vbfnoteX{#1}{#2}{\thisc@footnote}%
3909       \noexpand\led@reinit@index@fornote%
3910       \unexpanded{\advance\@edindex@fornote@m@ne}%
3911     }%
3912     \to\inserts@listR
3913     \global\advance\insert@countR \@ne%
3914   \else%
3915     \ifluatex
3916       \footnotelang@lua%
3917     \fi

```

```

3918 \ifundefined{xpg@main@language}%if polyglossia
3919 {}%
3920 {\footnotelang@poly}%
3921 \xright@appenditem{%
3922 \noexpand\led@set@index@fornote{#1}%
3923 \noexpand\prepare@edindex@fornote{\l@d@nums}%
3924 \unexpanded{\def\this@footnoteX@reading}{\the\csname footnote#1
@reading\endcsname}%
3925 \noexpand\vbfnoteX{#1}{#2}{\thisc@footnote}%
3926 \noexpand\led@reinit@index@fornote%
3927 \unexpanded{\advance\@edindex@fornote@m@ne}%
3928 }%
3929 \to\inserts@list
3930 \global\advance\insert@count \@ne%
3931 \fi
3932 \ignorespaces}
3933
3934 %

```

**\get@thisfootnoteX** The macro `\get@thisfootnote` command just saves the footnote number in the `\thisfootnote` macro, depending on the use of pairing environments.

```

3935 \newcommand{\get@thisfootnoteX}[1]{%
3936 \ifboolexpr{bool{l@dpairing} or bool{l@dprintingpages} or bool{
l@dprintingcolumns}}{%
3937 \protected@xdef\thisc@footnote{\the\csname footnote#1@reading\
endcsname}%
3938 }{%
3939 \protected@xdef\thisc@footnote{\the\csname c@footnote#1\endcsname}%
3940 }%
3941 }%
3942 %

```

**\vbfnoteX** This command calls the correct footnote-inserting commands.

```

3943 \newcommand{\vbfnoteX}[3]{%
3944 \get@fnmarkX{#1}{#3}%
3945 \@nameuse{regvfootnote#1}{#1}{#2}%
3946 }%
3947
3948 %

```

**\get@fnmarkX** This command gets the correct footnote number when typesetting parallel texts.

```

3949 \newcommand{\get@fnmarkX}[2]{%
3950 \ifboolexpr{bool{l@dpairing} or bool{l@dprintingpages} or bool{
l@dprintingcolumns}}{%
3951 {%
3952 \stepcounter{footnote#1@typeset}%
3953 \setcounter{footnote#1}{\value{footnote#1@typeset}}%

```

```

3954 \namedef{@thefnmark#1}{\csuse{thefootnote#1}}%
3955 \immediate\write\@mainaux{%
3956 \csgdef{footnote#1reading#2=typeset}{\the\csname c@footnote#1
@typeset\endcsname}%
3957 }%
3958 }%
3959 {%
3960 \setcounter{footnote#1}{#2}%
3961 \namedef{@thefnmark#1}{\csuse{thefootnote#1}}%
3962 }%
3963 }
3964 %
3965 %

```

```

\newcommand{\vnumfootnoteX}[2]{%
3967 \ifnumberedpar@
3968 \edtext{}{\normalbfnoteX{#1}{#2}}%
3969 \else
3970 \def\this@footnoteX@reading{\the\csname footnote#1@reading\endcsname}%
3971 \get@thisfootnoteX{#1}%
3972 \get@fnmarkX{#1}{\expandonce\thisc@footnote}%
3973 \@nameuse{regvfootnote#1}{#1}{#2}%
3974 \fi}
3975 %
3976 %

```

**arrangementX@normal** \arrangementX@normal{<series>} initialises the settings for the <series> footnotes. This should always be called for each series.

```

3977 \newcommand*{\arrangementX@normal}[1]{%
3978 \csgdef{series@display#1}{normal}
3979 \expandafter\let\csname footstart#1\endcsname=\normalfootstartX
3980 \@namedef{@footnotemark#1}{\normal@footnotemarkX{#1}}
3981 \@namedef{bodyfootmark#1}{\normalbodyfootmarkX{#1}}
3982 \expandafter\let\csname regvfootnote#1\endcsname=\normalvfootnoteX
3983 \expandafter\let\csname vfootnote#1\endcsname=\vnumfootnoteX
3984 \expandafter\let\csname footfmt#1\endcsname=\normalfootfmtX
3985 \@namedef{footfootmark#1}{\normalfootfootmarkX{#1}}
3986 \expandafter\let\csname footgroup#1\endcsname=\normalfootgroupX
3987 \expandafter\let\csname footnoterule#1\endcsname=\normalfootnoteruleX
3988 \count\csname footins#1\endcsname=1000
3989 \csxdef{default@footins#1}{1000}%Use to have note only for one side
3990 \dimen\csname footins#1\endcsname=\csuse{maxhnotesX@#1}
3991 \skip\csname footins#1\endcsname=\csuse{beforenotesX@#1}%
3992 \advance\skip\csname footins#1\endcsname by\csuse{afterruleX@#1}%
3993 %

```

Additions for minipages.

```

3994 \ifnoledgroup@else%
3995   \expandafter\let\csname mpvfootnote#1\endcsname=\mpnormalvfootnoteX
3996   \expandafter\let\csname mpfootgroup#1\endcsname=\mpnormalfootgroupX
3997   \count\csname mpfootins#1\endcsname=1000
3998   \dimen\csname mpfootins#1\endcsname=\csuse{maxhnotesX@#1}%
3999   \skip\csname mpfootins#1\endcsname=\csuse{beforenotesX@#1}%
4000   \advance\skip\csname mpfootins#1\endcsname by\csuse{afterruleX@#1}%
4001 \fi
4002 }
4003
4004 %

```

### 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 4005 \newcommand*\arrangementX@twocol}[1]{%
4006   \csgdef{series@displayX#1}{twocol}
4007   \expandafter\let\csname regvfootnote#1\endcsname=\twocolvfootnoteX
4008   \expandafter\let\csname footfmt#1\endcsname=\twocolfootfmtX
4009   \expandafter\let\csname footgroup#1\endcsname=\twocolfootgroupX
4010   \dimen\csname footins#1\endcsname=\csuse{maxhnotesX@#1}%
4011   \skip\csname footins#1\endcsname=\csuse{beforenotesX@#1}%
4012   \advance\skip\csname footins#1\endcsname by \csuse{afterruleX@#1}\relax%
4013   \twocolfootsetupX{#1}
4014   \ifnoledgroup@else%
4015     \expandafter\let\csname mpvfootnote#1\endcsname=\mpnormalvfootnoteX
4016     \expandafter\let\csname mpfootgroup#1\endcsname=\mptwocolfootgroupX
4017     \skip\csname mpfootins#1\endcsname=\csuse{beforenotesX@#1}%
4018     \advance\skip\csname mpfootins#1\endcsname by\csuse{afterruleX@#1}
4019     \mptwocolfootsetupX{#1}
4020   \fi%
4021 }
4022
4023 %

```

```

\twocolfootsetupX \twocolfootsetupX{<series>}
\mptwocolfootsetupX
4024 \newcommand*\twocolfootsetupX[1]{%
4025   \count\csname footins#1\endcsname 500
4026   \csxdef{default@footins#1}{500}%Use this to confine the notes to one
side only
4027   \multiply\dimen\csname footins#1\endcsname by \tw@}
4028 \newcommand*\mptwocolfootsetupX[1]{%
4029   \count\csname mpfootins#1\endcsname 500
4030   \multiply\dimen\csname mpfootins#1\endcsname by \tw@}
4031
4032 %

```



`\twocolvfootnoteX` `\twocolvfootnoteX{<series>}`

```

4033 \notbool{parapparatus@}{\newcommand*}{\newcommand}{\twocolvfootnoteX}[2]{%
4034   \csuse{beforeinsertingX@#1}%
4035   \insert\csname footins#1\endcsname\bgroup%
4036     \hsize=\expandafter\dimexpr\csuse{widthX@#1}\relax%
4037     \noindent\csuse{bhooknoteX@#1}%
4038     \csuse{notefontsizeX@#1}%
4039     \footssplitsskip%
4040     \spaceskip=\z@skip \xspaceskip=\z@skip%
4041     \@nameuse{footfmt#1}{#1}{#2}\egroup}
4042 %
4043 %

```

`\twocolfootfmtX` `\twocolfootfmtX{<series>}`

```

4044 \notbool{parapparatus@}{\newcommand*}{\newcommand}{\twocolfootfmtX}[2]{%
4045   \protected@edef\@currentlabel{%
4046     \@nameuse{@thefnmark#1}%
4047   }%
4048   \normal@pars%
4049   \hangindent=\csuse{hangindentX@#1}%
4050   \everypar{\hangindent=\csuse{hangindentX@#1}}%
4051   \hspace\csuse{hsizetwocolX@#1}%
4052   \nottoggle{parindentX@#1}{\parindent=\z@}{}%
4053   \tolerance=5000\relax%
4054   \par%
4055   \@tempdima=\parindent%
4056   \csuse{colalignX@#1}%
4057   \parindent=\@tempdima%
4058   {\hspace{\parindent}%
4059    \csuse{notenunfontX@#1}\wrapped@footfootmarkX{#1}\strut%
4060    \csuse{wrapcontentX@#1}{#2}%
4061    \strut\par}%
4062   \allowbreak%
4063 }%
4064 %
4065 %

```

`\twocolfootgroupX` `\twocolfootgroupX{<series>}`

`\mptwocolfootgroupX`

```

4066 \newcommand*{\twocolfootgroupX}[1]{\csuse{bhookgroupX@#1}\csuse{
notefontsizeX@#1}
4067   \splittopskip=\ht\strutbox
4068   \expandafter
4069   \rigidbalanceX\csname footins#1\endcsname \tw@ \splittopskip}}
4070 %
4071 \newcommand*{\mptwocolfootgroupX}[1]{%
4072   \vskip\skip\@nameuse{mpfootins#1}
4073   \ifl@dpairing\ifparledgroup%

```

```

4074 \leavevmode\marks\parledgroup@{begin}%
4075 \marks\parledgroup@series{#1}%
4076 \marks\parledgroup@type{footnoteX}%
4077 \fi\fi\normalcolor
4078 \ifparledgroup%
4079 \ifl@dpairing%
4080 \else%
4081 \setnoteswidthliketwocolumnsX@{#1}%
4082 \setnotesXpositionliketwocolumns@{#1}%
4083 \print@footnoteXrule{#1}%
4084 \fi%
4085 \else%
4086 \setnoteswidthliketwocolumnsX@{#1}%
4087 \setnotesXpositionliketwocolumns@{#1}%
4088 \print@footnoteXrule{#1}%
4089 \fi%
4090 \csuse{bhookgroupX@#1}%
4091 \splittopskip=\ht\strutbox
4092 \expandafter
4093 \rigidbalanceX\csname mpfootins#1\endcsname \tw@ \splittopskip}}
4094
4095 %

```

#### 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% \newcommand*{\arrangementX@threecol}[1]{%
4097 \csgdef{series@displayX#1}{threecol}
4098 \expandafter\let\csname regvfootnote#1\endcsname=\threecolvfootnoteX
4099 \expandafter\let\csname footfmt#1\endcsname=\threecolfootfmtX
4100 \expandafter\let\csname footgroup#1\endcsname=\threecolfootgroupX
4101 \dimen\csname footins#1\endcsname=\csuse{maxhnotesX@#1}%
4102 \skip\csname footins#1\endcsname=\csuse{beforenotesX@#1}%
4103 \advance\skip\csname footins#1\endcsname by \csuse{afterruleX@#1}\relax%
4104 \threecolfootsetupX{#1}
4105 \ifnoledgroup@else%
4106 \expandafter\let\csname mpvfootnote#1\endcsname=\mpnormalvfootnoteX
4107 \expandafter\let\csname mpfootgroup#1\endcsname=\mpthreecolfootgroupX
4108 \skip\csname mpfootins#1\endcsname=\csuse{beforenotesX@#1}%
4109 \advance\skip\csname mpfootins#1\endcsname by \csuse{afterruleX@#1}
4110 \mpthreecolfootsetupX{#1}
4111 \fi%
4112 }
4113
4114 %

```

```

\threecolfootsetupX \threecolfootsetupX{<series>}
\mpthreecolfootsetupX

```

```

4115 \newcommand*{\threecolfootsetupX}[1]{%
4116   \count\csname footins#1\endcsname 333
4117   \csxdef{default@footins#1}{333}%Use this to confine the notes to one
side only
4118   \multiply\dimen\csname footins#1\endcsname by \thr@@
4119 \newcommand*{\mpthreecolfootsetupX}[1]{%
4120   \count\csname mpfootins#1\endcsname 333
4121   \multiply\dimen\csname mpfootins#1\endcsname by \thr@@
4122
4123 %

```

`\threecolvfootnoteX` `\threecolvfootnoteX{<series>}{<text>}`

```

4124 \notbool{parapparatus@}{\newcommand*}{\newcommand}{\threecolvfootnoteX}[2]{%
%
4125   \csuse{beforeinsertingX@#1}%
4126   \insert\csname footins#1\endcsname\bgroup%
4127     \hspace=\expandafter\dimexpr\csuse{widthX@#1}\relax%
4128     \noindent\csuse{bhooknoteX@#1}%
4129     \csuse{notefontsizeX@#1}%
4130     \footsplitskips%
4131     \@nameuse{footfmt#1}{#1}{#2}\egroup
4132
4133 %

```

`\threecolfootfmtX` `\threecolfootfmtX{<series>}`

```

4134 \notbool{parapparatus@}{\newcommand*}{\newcommand}{\threecolfootfmtX}[2]{%
4135   \protected@edef\@currentlabel{%
4136     \@nameuse{@thefnmark#1}%
4137   }%
4138   \hangindent=\csuse{hangindentX@#1}%
4139   \everypar{\hangindent=\csuse{hangindentX@#1}}%
4140   \normal@pars%
4141   \hspace \csuse{hsizethreecolX@#1}%
4142   \nottoggle{parindentX@#1}{\parindent=\z@}{}%
4143   \tolerance=5000\relax%
4144   \@tempdima=\parindent%
4145   \csuse{colalignX@#1}%
4146   \parindent=\@tempdima%
4147   {\hspace{\parindent}%
4148     \csuse{notenumfontX@#1}\wrapped@footfootmarkX{#1}\strut%
4149     \csuse{wrapcontentX@#1}{#2}%
4150     \strut\par}\allowbreak
4151
4152 %

```

`\threecolfootgroupX` `\threecolfootgroupX{<series>}`  
`\mpthreecolfootgroupX`

```

4153 \newcommand*\threecolfootgroupX}[1]{\csuse{bhookgroupX@#1}\csuse{
notefontsizeX@#1}
4154 \splittopskip=\ht\strutbox
4155 \expandafter
4156 \rigidbalanceX\csname footins#1\endcsname \thr@@ \splittopskip}}
4157
4158 \newcommand*\mpthreecolfootgroupX}[1]{\%
4159 \vskip\skip\@nameuse{mpfootins#1}
4160 \ifl@dpairing\ifparledgroup
4161 \leavevmode\marks\parledgroup@{begin}%
4162 \marks\parledgroup@series{#1}%
4163 \marks\parledgroup@type{footnoteX}%
4164 \fi\fi\normalcolor
4165 \ifparledgroup%
4166 \ifl@dpairing%
4167 \else%
4168 \setnoteswidthliketwocolumnsX@{#1}%
4169 \setnotesXpositionliketwocolumns@{#1}%
4170 \print@footnoteXrule{#1}%
4171 \fi%
4172 \else%
4173 \setnoteswidthliketwocolumnsX@{#1}%
4174 \setnotesXpositionliketwocolumns@{#1}%
4175 \print@footnoteXrule{#1}%
4176 \fi%
4177 \csuse{bhookgroupX@#1}%
4178 \splittopskip=\ht\strutbox
4179 \expandafter
4180 \rigidbalanceX\csname mpfootins#1\endcsname \thr@@ \splittopskip}}
4181
4182 %

```

### XIII.4.5 Paragraphed footnotes

The following macros set footnotes as one paragraph.

`\arrangementX@threecol` `\footparagraphX{<series>}`

```

4183 \newcommand*\arrangementX@paragraph}[1]{\%
4184 \csgdef{series@displayX#1}{paragraph}%
4185 \expandafter\let\csname footstart#1\endcsname=\parafootstartX
4186 \expandafter\let\csname regvfootnote#1\endcsname=\para@vfootnoteX
4187 \expandafter\let\csname footfmt#1\endcsname=\parafootfmtX
4188 \expandafter\let\csname footgroup#1\endcsname=\para@footgroupX
4189 \expandafter\let\csname footnoterule#1\endcsname=\normalfootnoteruleX
4190 \count\csname footins#1\endcsname=1000
4191 \csxdef{default@footins#1}{1000}%Use this to confine the notes to one
side only
4192 \dimen\csname footins#1\endcsname=\csuse{maxhnotesX@#1}
4193 \skip\csname footins#1\endcsname=\csuse{beforenotesX@#1}%

```

```

4194 \advance\skip\csname footins#1\endcsname by\csuse{afterruleX@#1}%
4195 \para@footsetupX{#1}
4196 \ifnoledgroup@ \else
4197   \expandafter\let\csname mpvfootnote#1\endcsname=\mppara@vfootnoteX
4198   \expandafter\let\csname mpfootgroup#1\endcsname=\mppara@footgroupX
4199   \count\csname mpfootins#1\endcsname=1000
4200   \dimen\csname mpfootins#1\endcsname=\csuse{maxhnotesX@#1}
4201   \skip\csname mpfootins#1\endcsname=\csuse{beforenotesX@#1}%
4202   \advance\skip\csname mpfootins#1\endcsname by\csuse{afterruleX@#1}%
4203 \fi
4204 }
4205
4206 %

```

`\para@footsetupX` `\para@footsetupX{<series>}`

```

4207 \newcommand*{\para@footsetupX}[1]{\csuse{bhookgroupX@#1}\csuse{
notefontsizeX@#1}
4208 \setnoteswidthliketwocolumnsX@{#1}%
4209 \ifcsempy{widthX@#1}%
4210   {}%
4211   {\columnwidth=\expandafter\dimexpr\csuse{widthX@#1}\relax}%
4212 \dimen0=\baselineskip
4213 \multiply\dimen0 by 1024
4214 \divide\dimen0 by \columnwidth \multiply\dimen0 by \footfudgefiddle\relax
4215 %
4216 \expandafter
4217 \xdef\csname footfudgefactor#1\endcsname{%
\expandafter\strip@pt\dimen0 }}
4218
4219 %

```

`\parafootstartX` `\parafootstartX{<series>}`

```

4220 \newcommand*{\parafootstartX}[1]{%
4221   \ifdimequal{0pt}{\prenotesX@}{}%
4222   {%
4223     \iftoggle{prenotesX@}{%
4224       \togglefalse{prenotesX@}%
4225       \skip\csname footins#1\endcsname=%
4226       \glueexpr\csuse{prenotesX@}+\csuse{afterruleX@#1}\relax%
4227     }%
4228     {}%
4229   }%
4230 \leftskip=\z@
4231 \rightskip=\z@
4232 \nottoggle{parindentX@#1}{\parindent=\z@}{}%
4233 \vskip\skip\@nameuse{footins#1}%
4234 \setnoteswidthliketwocolumnsX@{#1}%
4235 \setnotesXpositionliketwocolumns@{#1}%

```

```

4236 \print@footnoteXrule{#1}%
4237 }
4238
4239 %

```

```

\para@vfootnoteX \para@vfootnoteX{<series>}{<text>}
\mppara@vfootnoteX
4240 \newcommand*{\para@vfootnoteX}[2]{%
4241 \csuse{beforeinsertingX@#1}%
4242 \insert\csname footins#1\endcsname%
4243 \bgroup
4244 \csuse{notefontsizeX@#1}
4245 \footsplitskips
4246 \setbox0=\vbox{\hsize=\maxdimen%
4247 \let\bidir@RTL@everypar\@empty%
4248 \noindent\csuse{bhooknoteX@#1}%
4249 \@nameuse{footfmt#1}{#1}{#2}}%
4250 \setbox0=\hbox{\unvxhX{0}{#1}}%
4251 \dp0=\z@
4252 \ht0=\csname footfudgefactor#1\endcsname\wd0
4253 \box0
4254 \penalty0
4255 \egroup}
4256 \newcommand*{\mppara@vfootnoteX}[2]{%
4257 \get@thisfootnoteX{#1}%
4258 \get@fnmarkX{#1}{\thisc@footnote}%
4259 \edef\this@footnoteX@reading{\the\csname footnote#1@reading\endcsname}%
4260 \global\setbox\@nameuse{mpfootins#1}\vbox{%
4261 \unvbox\@nameuse{mpfootins#1}
4262 \csuse{notefontsizeX@#1}
4263 \footsplitskips
4264 \setbox0=\vbox{\hsize=\maxdimen%
4265 \let\bidir@RTL@everypar\@empty%
4266 \noindent\color@begingroup%
4267 \csuse{bhooknoteX@#1}%
4268 \@nameuse{footfmt#1}{#1}{#2}\color@endgroup}%
4269 \setbox0=\hbox{\unvxhX{0}{#1}}%
4270 \dp0=\z@
4271 \ht0=\csname footfudgefactor#1\endcsname\wd0
4272 \box0
4273 \penalty0}}
4274
4275 %

```

```

\unvxhX76 \newcommand*{\unvxhX}[2]{% 2th is optional for retro-compatibility
4277 \setbox0=\vbox{\unvbox#1%
4278 \global\setbox1=\lastbox}%
4279 \unhbox1
4280 \unskip % remove \rightskip,

```

```

4281 \unskip           % remove \parfillskip,
4282 \unpenalty        % remove \penalty of 10000,
4283 \hskip\csuse{afternoteX@#2}%
4284 \relax}% but add the glue to go between the notes
4285
4286 %

```

`\parafootfmtX` `\parafootfmtX{<series>}`

```

4287 \newcommand*{\parafootfmtX}[2]{%
4288   \protected@edef\@currentlabel{%
4289     \@nameuse{@thefnmark#1}%
4290   }%
4291   \insertparafootsepX{#1}%
4292   \ledsetnormalparstuff@common%
4293   {\csuse{notenunfontX@#1}%
4294    \csuse{notenunfontX@#1}%
4295    \wrapped@footfootmarkX{#1}%
4296    \strut%
4297    \csuse{wrapcontentX@#1}{#2}%
4298    \penalty-10}}
4299
4300 %

```

`\para@footgroupX` `\para@footgroupX{<series>}`

`\mppara@footgroupX`

```

4301 \newcommand*{\para@footgroupX}[1]{%
4302   \hsize=\expandafter\dimexpr\csuse{widthX@#1}\relax%
4303   \unvbox\curname footins#1\endcsname
4304   \ifcsstring{raggedX@#1}{L}{\RaggedLeft}{}%
4305   \ifcsstring{raggedX@#1}{R}{\RaggedRight}{}%
4306   \makehboxofhboxes
4307   \setbox0=\hbox{\unhbox0 \removehboxes}%
4308   \csuse{bhookgroupX@#1}
4309   \csuse{notefontsizeX@#1}
4310   \unhbox0\par}
4311
4312 \newcommand*{\mppara@footgroupX}[1]{%
4313   \setnoteswidthliketwocolumnsX@{#1}%
4314   \vskip\skip\@nameuse{mpfootins#1}
4315   \ifl@dpairing\ifparledgroup
4316     \leavevmode%
4317     \leavevmode\marks\parledgroup@{begin}%
4318     \marks\parledgroup@series{#1}%
4319     \marks\parledgroup@type{footnoteX}%
4320   \fi\fi\normalcolor
4321   \ifparledgroup%
4322     \ifl@dpairing%
4323     \else%
4324       \setnoteswidthliketwocolumnsX@{#1}%

```

```

4325 \setnotesXpositionliketwocolumns@{#1}%
4326 \print@footnoteXrule{#1}%
4327 \fi%
4328 \else%
4329 \setnoteswidthliketwocolumnsX@{#1}%
4330 \setnotesXpositionliketwocolumns@{#1}%
4331 \print@footnoteXrule{#1}%
4332 \fi%
4333 \unvbox\csname mpfootins#1\endcsname
4334 \ifcsstring{raggedX@#1}{L}{\RaggedLeft}{}%
4335 \ifcsstring{raggedX@#1}{R}{\RaggedRight}{}%
4336 \makehboxofhboxes
4337 \setbox0=\hbox{\unhbox0 \removehboxes}%
4338 \csuse{hookgroupX@#1}%
4339 \csuse{notefontsizeX@#1}%
4340 \nottoggle{parindentX@#1}{\parindent=\z@}{}%
4341 \unhbox0\par}}
4342
4343 %

```

**Insertion of the footnotes separator** The command `\insertparafootsepX{<series>}` must be called at the beginning of `\parafootftmX`.

```

\insertparafootsepX%44 \newcommand{\insertparafootsepX}[1]{%
4345 \ifledRcol%
4346 \ifnumequal{\csuse{prevpage#1@numR}}{\page@numR}%
4347 {\csuse{Xparafootsep@#1}}%
4348 {}%
4349 \global\csname prevpage#1@numR\endcsname=\page@numR%
4350 \else%
4351 \ifnumequal{\csuse{prevpage#1@num}}{\page@num}%
4352 {\csuse{Xparafootsep@#1}}%
4353 {}%
4354 \global\csname prevpage#1@num\endcsname=\page@num%
4355 \fi%
4356 }
4357 %

```

### XIII.5 Wrapping footnote marks in hyperlink

`\wrapped@footfootmarkX` `\wrapped@footfootmarkX` prints the footnote mark of the footpage, wrapped in `hyperref` package's commands, if needed.

```

4358 \newcommand{\wrapped@footfootmarkX}[1]{%
4359 \ifdefined\hypertarget%
4360 \hyperlink%
4361 {\@bodyfootmark#1@this@footnoteX@reading}%
4362 {\@nameuse{footfootmark#1}}%

```



```

4363 \Hy@raisedlink{%
4364 \hypertarget%
4365 {@footnotemark#1@\this@footnoteX@reading}%
4366 {}%
4367 }%
4368 \else%
4369 \@nameuse{footfootmark#1}%
4370 \fi%
4371 }%
4372 %

```

`\wrapped@bodyfootmarkX` `\wrapped@bodyfootmarkX` prints the footnote mark of the text body, wrapped in `hyperref` package’s commands, if needed.

```

4373 \newcommand{\wrapped@bodyfootmarkX}[1]{%
4374 \ifdefined\hypertarget%
4375 \hyperlink%
4376 {@footnotemark#1@\expandafter\the\csname footnote#1@reading\
endcsname}%
4377 {@\nameuse{bodyfootmark#1}}}%
4378 \Hy@raisedlink{%
4379 \hypertarget%
4380 {@bodyfootmark#1@\expandafter\the\csname footnote#1@reading\
endcsname}%
4381 {}%
4382 }%
4383 \else%
4384 \@nameuse{bodyfootmark#1}%
4385 \fi%
4386 }%
4387 %

```

## 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 `LuaTeX`.

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.

```

\ledsetnormalparstuff@common% \newcommand*{\ledsetnormalparstuff@common}{%
\Xledsetnormalparstuff% \ifluatex%
\ledsetnormalparstuff% \ifdefstring{\footnote@luatextextdir}{TLT}{}%

```

```

4391 {\textdir\footnote@luatexttextdir}%
4392 \pardir\footnote@luatexpardir%
4393 \fi%
4394 \csuse{\csuse{footnote@dir}}%
4395 \normal@pars%
4396 \parfillskip \z@ \@plus 1fil}%
4397
4398 \newcommand*{\Xledsetnormalparstuff}[1]{%
4399 \ledsetnormalparstuff@common%
4400 \nottoggle{Xparindent@#1}{\parindent=\z@}{\hspace{\parindent}}%
4401 }%
4402
4403 \newcommand*{\ledsetnormalparstuffX}[1]{%
4404 \ledsetnormalparstuff@common%
4405 \nottoggle{parindentX@#1}{\parindent=\z@}{\hspace{\parindent}}%
4406 }%
4407 %

```

## 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. They set, if the option is enabled, the `\hsize`. They are also called at the on the setup for paragraphed notes.

`\setXnoteswidthliketwocolumns@`

`\setnoteswidthliketwocolumnsX@`

```

4408
4409 \newdimen\old@hsize%
4410 \AtBeginDocument{\old@hsize=\hsize}%
4411
4412 \newcommand{\setXnoteswidthliketwocolumns@}[1]{%
4413 \global\let\hsize@fornote=\hsize%
4414 \global\old@hsize=\hsize%
4415 \let\old@columnwidth=\columnwidth%
4416 \iftoggle{Xnoteswidthliketwocolumns@#1}%
4417 {%
4418 \csuse{setwidthliketwocolumns@\columns@position}%
4419 \global\let\hsize@fornote=\hsize%
4420 }%
4421 }%
4422 \let\hsize=\hsize@fornote%
4423 \let\columnwidth=\old@columnwidth%
4424 }%
4425
4426 \newcommand{\setnoteswidthliketwocolumnsX@}[1]{%
4427 \global\let\hsize@fornote=\hsize%
4428 \global\old@hsize=\hsize%

```

```

4429 \let\old@columnwidth=\columnwidth%
4430 \iftoggle{noteswidthliketwocolumnsX@#1}{%
4431   {%
4432     \csuse{setwidthliketwocolumns@\columns@position}%
4433     \global\let\hsize@fornote=\hsize%
4434   }%
4435   {%
4436     \let\hsize=\hsize@fornote%
4437     \let\columnwidth=\old@columnwidth%
4438   }%
4439
4440 %

```

`\setnotespositionliketwocolumns@` These two commands set the position of the critical / familiar footnotes, depending on the hooks `Xnoteswidthliketwocolumns` and `noteswidthliketwocolumnsX`. They call commands which are defined only in `reledpar`, because this feature has no sense without `reledpar`.

```

4441 \newcommand{\setXnotespositionliketwocolumns@}[1]{%
4442   \iftoggle{Xnoteswidthliketwocolumns@#1}{%
4443     \csuse{setnotespositionliketwocolumns@\columns@position}%
4444   }{%
4445   }%
4446
4447 \newcommand{\setnotesXpositionliketwocolumns@}[1]{%
4448   \iftoggle{noteswidthliketwocolumnsX@#1}{%
4449     \csuse{setnotespositionliketwocolumns@\columns@position}%
4450   }{%
4451   }%
4452
4453 %

```

## 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.

```

4454 \def\@fnpos{familiar-critical}
4455 \def\@mpfnpos{critical-familiar}
4456 \newcommand{\fnpos}[1]{\xdef\@fnpos{#1}}
4457 \newcommand{\mpfnpos}[1]{\xdef\@mpfnpos{#1}}
4458 %

```

## 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\newcommand{\print@Xfootnoterule}[1]{%
\print@footnotexrule\vskip-\csuse{Xafterterrule@#1}%Because count in \dimen\csuse{#1footins}
4461 \nointerlineskip%
4462 \moveleft-\leftskip\ vbox{\csuse{#1footnoterule}}%
4463 \nointerlineskip%
4464 \vskip\csuse{Xafterterrule@#1}%
4465 }%
4466
4467 \newcommand{\print@footnotexrule}[1]{%
4468 \vskip-\csuse{afterterruleX@#1}%Because count in \dimen\csuse{footins#1}
4469 \nointerlineskip%
4470 \moveleft-\leftskip\ vbox{\csuse{footnoterule#1}}%
4471 \nointerlineskip%
4472 \vskip\csuse{afterterruleX@#1}%
4473 }%
4474
4475 %

```

## 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@Xprenotes 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

`\Xprenotes@` If user redefines `\Xprenotes@`, via `\Xprenotes` to a value greater than 0 pt, this skip will be added before first series notes instead of the notes skip.

```

4476 \newtoggle{Xprenotes@}%
4477 \toggletrue{Xprenotes@}%
4478 \newcommand{Xprenotes@}{Opt}%
4479 \newcommand*{Xprenotes}[1]{\renewcommand{Xprenotes@}{#1}}%
4480 \newcommand{preXnotes}[1]{\led@warning@preXnotes@deprecatedXprenotes{#1}}
    %For compatibility
4481 %

```

The same, but for familiar footnotes.

```

\Xprenotes@ \newtoggle{prenotesX@}
\Xprenotes@ \toggletrue{prenotesX@}
4484 \newcommand{prenotesX@}{Opt}
4485 \newcommand*{prenotesX}[1]{\renewcommand{prenotesX@}{#1}}
4486 %

```

### XVIII.0.3 Internal commands

```

firstXseries@ \gdef\firstXseries@{}
prepare@Xprenotes@ \newcommand{prepare@Xprenotes}[1]{%
4489 \ifdimequal{Opt}{Xprenotes@}%
4490 {}%
4491 {}%
4492 \IfStrEq{firstXseries@}{}%
4493 \global\skip\csuse{#1footins}=\Xprenotes@%
4494 \global\advance\skip\csname #1footins\endcsname by\csuse{Xafterrule@
#1}%
4495 \gdef\firstXseries@{#1}%
4496 }%
4497 {%
4498 \ifseriesbefore{#1}{firstXseries@}%
4499 {%
4500 \global\skip\csuse{#1footins}=\csuse{Xbeforenotes@firstXseries@}%
4501 \global\advance\skip\csname #1footins\endcsname by\csuse{Xafterrule@
#1}%
4502 \gdef\firstXseries@{#1}%
4503 }%
4504 {}%

```

```

4505 }%
4506 }%
4507 }
4508 %

```

The same thing is required for familiar notes and `\prenotesX`.

```

firstseriesX@09 \gdef\firstseriesX@{}
prepare@prenotesX10 \newcommand{\prepare@prenotesX}[1]{%
4511 \ifdimequal{0pt}{\prenotesX@}%
4512 {}%
4513 {%
4514 \IfStrEq{\firstseriesX@}{-}{%
4515 \global\skip\csuse{footins#1}=\prenotesX@%
4516 \global\advance\skip\csname footins#1\endcsname by\csuse{afterruleX@
#1}%
4517 \gdef\firstseriesX@{#1}%
4518 }%
4519 {%
4520 \ifseriesbefore{#1}{\firstseriesX@}%
4521 {%
4522 \global\skip\csuse{footins#1}=\csuse{beforenotesX@firstseriesX@}%
4523 \global\advance\skip\csname footins#1\endcsname by\csuse{afterruleX@
#1}%
4524 \gdef\firstXseries@{#1}%
4525 }%
4526 {}%
4527 }%
4528 }%
4529 }
4530 %

```

## XIX Endnotes

First, check the `noend` option.

```

4531 \ifbool{noend@}{-}{%Used instead of \ifnoend@ to prevent expansion problem
4532 %

```

### XIX.1 Internal commands

`\l@dend@open` and `\l@dend@close` are the macros that are used to open and close the endnote file. Note that all our writing to this file is `\immediate`: all page and line numbers for the endnotes are generated by the same mechanism we use for the footnotes, so that there is no need to defer any writing to catch information from the output routine. The argument of these two command is the series letter.

```

4533 \newcommand{\l@dend@open}[1]{%
4534   \global\booltrue{l@dend@#1}%
4535   \expandafter\immediate%
4536   \expandafter\openout%
4537   \csname l@d@#1end\endcsname%
4538   =\l@auxdir\jobname.#1end\relax%
4539 }%
4540 \newcommand{\l@dend@close}[1]{%
4541   \global\boolfalse{l@dend@#1}%
4542   \expandafter\immediate%
4543   \expandafter\closeout\csname l@d@#1end\endcsname%
4544 }%
4545 %
4546 %

```

**\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.

```

4547 \newcommand{\l@dend@stuff}{%
4548   \def\do##1{%
4549     \ifbool{l@dend@##1}{}%
4550     {\l@dend@open{##1}}%
4551     \expandafter\immediate\expandafter\write\csname l@d@##1end\endcsname{\
string\l@d@section{\the\section@num}\@percentchar}%
4552   }%
4553   \dolistloop{@series}%
4554 }%
4555 %
4556 %

```

**\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.
- #6 Side (L or R).
- #7 Label for cross-referencing.

```

4557 \global\newbool{parapparatus@#4}\def\endprint#1#2#3#4#5#6#7{
4558   \csuse{Xendbhooknote@#4}%
4559   \csuse{Xendnotefontsize@#4}%
4560   \hangindent=\csuse{Xendhangindent@#4}%
4561   \ifXendinsertsep%
4562     \hskip\csuse{Xendafternote@#4}\relax%
4563     \csuse{Xendsep@#4}%
4564   \else%
4565     \iftoggle{Xendparagraph@#4}%
4566       {\global\Xendinsertsep@true}%
4567       {}%
4568   \fi%
4569   \xdef\@currentseries{#4}%
4570   \def\do##1{%
4571     \setkeys[mac]{truefootnoteoption}{##1}%
4572   }%
4573   \notblank{#5}{\docsvlist{#5}}{}%
4574   \IfStrEq{#6}{R}{\ledRcol@true}{}%
4575   \def\@this@crossref@start{#7:start}%
4576   \def\@this@crossref@end{#7:end}%
4577   \printlineendnote{#1}{#4}%
4578   \IfStrEq{#6}{R}{\ledRcol@false}{}%
4579   \undef\@this@crossref@start%
4580   \undef\@this@crossref@end%
4581   \nottoggle{Xendlemmadisablefontselection@#4}%
4582     {\select@lemmafont#1}{}%
4583   {}%
4584   \bgroup%
4585     \csuse{Xendlemmafont@#4}%
4586     \csuse{Xendwraplemma@#4}{#2}%
4587   \egroup%
4588   \ifboolexpr{
4589     togl {nosep@}%
4590     or test{\ifcempty{Xendlemmaseparator@#4}}%
4591   }%
4592     {\hskip\csuse{Xendinplaceoflemmaseparator@#4}\relax}%
4593     {\nobreak%
4594       \hskip\csuse{Xendbeforelemmaseparator@#4}%
4595       \csuse{Xendlemmaseparator@#4}%
4596       \hskip\csuse{Xendafterlemmaseparator@#4}%
4597       \relax%
4598     }%
4599   \csuse{Xendwrapcontent@#4}{#3}%
4600   \nottoggle{Xendparagraph@#4}{\par}{}%
4601   \def\do##1{%
4602     \setkeys[mac]{falsefootnoteoption}{##1}%
4603   }%
4604   \notblank{#5}{\docsvlist{#5}}{}%
4605 }%
4606

```



```

4607 \let\l@d@section=\@gobble
4608
4609 %

```

**\printlineendnote** This macro controls, in endnote, 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.

```

4610 \newcommand{\printlineendnote}[2]{%
4611   \l@dp@rsefootspec#1|{%
4612     \iftoggle{Xendnumberonlyfirstintwolines@#2}{%
4613       \xdef\lineinfo@{\l@d@p@rsestartpage - \l@d@p@rsestartline - \
4614         \l@d@p@rsestartsub - \l@d@p@rseendpage - \l@d@p@rseendline - \
4615         \l@d@p@rseendsub}%
4616       }%
4617       {%
4618         \xdef\lineinfo@{\l@d@p@rsestartpage - \l@d@p@rsestartline - \
4619         \l@d@p@rsestartsub}%
4620       }%
4621     }%
4622     \ifboolexpr{%
4623       togl {nonum@}%
4624       or togl {Xendnonumber@#2}}%
4625     {%
4626       \iftoggle{Xendnumberonlyfirstinline@#2}%
4627       {\ifcsdef{prevendline#2}%
4628         {\ifcsequal{prevendline#2}{\lineinfo@}%
4629         {%
4630           \csuse{Xendbhookinplaceofnumber@#2}%
4631           \ifcsequal{Xendsymmlinenumber@#2}%
4632             {\hspace{\csuse{Xendinplaceofnumber@#2}}}%
4633             {\printsymmlineendnotearea{#2}}}%
4634           \csuse{Xendahookinplaceofnumber@#2}%
4635         }%
4636         {\printlineendnotearea{#1}{#2}}}%
4637       {\printlineendnotearea{#1}{#2}}}%We keep every time line
4638     \csxdef{prevendline#2}{\lineinfo@}%
4639   }%
4640 }%
4641 %

```

```

\printsymmlineendnotearea \newcommand{\printsymmlineendnotearea}[1]{%
4643   \hspace{\csuse{Xendbeforemlinenumber@#1}}%
4644   \csuse{Xendnotenumfont@#1}%
4645   \ifdimequal{\csuse{Xendboxsymmlinenumber@#1}}{\z@}%
4646   {\csuse{Xendsymmlinenumber@#1}}%

```

```

4647     {\hbox to \csuse{Xendboxsymlinenumber@#1}%
4648       {\csuse{Xendsymlinenumber@#1}\hfill}%
4649     }%
4650     \hspace{\csuse{Xendaftersymlinenumber@#1}}%
4651   }%
4652   %

```

`\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.)

```

4653 \newcommand{\printlineendnotearea}[2]{%
4654   \csuse{Xendbhooklinenumber@#2}%
4655   \hspace{\csuse{Xendbeforenumber@#2}}%
4656   \bgroup%
4657   \csuse{Xendnotenumfont@#2}%
4658   \ifdimequal{\csuse{Xendboxlinenumber@#2}}{0pt}%
4659     {\printendlines#1||\ifledRcol@\@Rlineflag\fi}%
4660     {\leavevmode%
4661       \hbox to \csuse{Xendboxlinenumber@#2}%
4662       {%
4663         \IfSubStr{RC}{\csuse{Xendboxlinenumberalign@#2}}{\hfill}}%
4664         \printendlines#1||\ifledRcol@\@Rlineflag\fi%
4665         \IfSubStr{LC}{\csuse{Xendboxlinenumberalign@#2}}{\hfill}}%
4666       }%
4667   \egroup%
4668   \hspace{\csuse{Xendafternumber@#2}}%
4669   \csuse{Xendahooklinenumber@#2}%
4670 }%
4671 %

```

## XIX.2 User level commands

### XIX.2.1 Inserting contents to endnotes

The `\Xendnotes` commands are defined above, when defining apparatus commands by series. Here, we define only `\toendnotes` command not specific to a series, in order to insert arbitrary code. The regular version writes an unexpanded argument, while the regular version writes a once-expanded argument.

```

\toendnotes* \newcommandx{\toendnotes}[2][1,usedefault]{%
\toendnotes* \ifboolexpr{bool{numbering} or bool{numberingR}}{%
4674   \def\do##1{%
4675     \expandafter\immediate\expandafter\write\csname l@d@##1end\endcsname%
4676     {\unexpanded{##1}\@percentchar}%
4677   }%
4678   \ifstrempy{#1}%
4679     {\dolistloop{\@series}}%

```

```

4680     {\docsvlist{#1}}%
4681   }{\led@err@toendnotes@outsidenumbering}%
4682 }%
4683 \WithSuffix\newcommandx\toendnotes*[2][1,usedefault]{%
4684   \ifboolexpr{bool{numbering} or bool{numberingR}}{%
4685     \def\do##1{%
4686       \expandafter\immediate\expandafter\write\csname l@d@##1end\endcsname%
4687       {#2\@percentchar}%
4688     }%
4689     \ifstrempy{#1}%
4690       {\dolistloop{\@series}}%
4691       {\docsvlist{#1}}%
4692   }{\led@err@toendnotes@outsidenumbering}%
4693 }%
4694 %

```

### XIX.2.2 Printing endnotes

**\doendnotes** **\ifXendinsertsep@** **\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.

```

4695 \newif\ifXendinsertsep@%
4696 \newcommand*{\doendnotes}[1]{%
4697   \l@dend@close{#1}%
4698   \begingroup
4699     \csxdef{prevpagenum@#1}{}%
4700     \csxdef{prevpagerange@#1}{}%
4701     \makeatletter
4702     \expandafter\let\csname #1end\endcsname=\endprint
4703     \input\l@auxdir\jobname.#1end%
4704     \global\Xendinsertsep@false%
4705   \endgroup}
4706 %

```

**\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.

```

4707 \newcommand*{\doendnotesbysection}[1]{%
4708   \l@dend@close{#1}%
4709   \csxdef{prevpagenum@#1}{}%
4710   \csxdef{prevpagerange@#1}{}%
4711   \global\expandafter\advance\csname #1end@bysection\endcsname by 1%
4712   \begingroup%
4713     \makeatletter%
4714     \def\l@d@section##1{%

```

```

4715     \ifnumequal{##1}{\csname #1end@bysection\endcsname}%
4716     {\cslet{#1end}{\endprint}}}%
4717     {\cslet{#1end}{\@gobbleseven}}}%
4718 }%
4719 \input\l@auxdir\jobname.#1end%
4720 \global\Xendinsertsep@false%
4721 \endgroup%
4722 }%
4723 %

```

We close now the conditional period, which depends on `\ifnoend@`, because the following commands can be used by other commands than those specific to endnotes.

```

4724 }%
4725 %

```

**`\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.

```

4726 \newcommand*{\setprintendlines}[6]{%
4727   \l@d@pnumfalse \l@d@dashfalse
4728   \ifnum#4=#1 \else
4729     \l@d@pnumtrue
4730     \l@d@dashtrue
4731   \fi
4732 %

```

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.

```

4733   \ifl@d@pnum \l@d@elintrue \else \l@d@elinfalse \fi
4734   \ifnum#2=#5 \else
4735     \l@d@elintrue
4736     \l@d@dashtrue
4737   \fi
4738 %

```

We print the starting sub-line if it is nonzero.

```

4739   \l@d@ssubfalse
4740   \ifnum#3=0 \else
4741     \l@d@ssubtrue
4742   \fi
4743 %

```

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.

```

4744 \l@d@eslfalse
4745 \ifnum#6=0 \else
4746   \ifnum#6=#3
4747     \ifl@d@elin \l@d@esltrue \else \l@d@eslfalse \fi
4748   \else
4749     \l@d@esltrue
4750     \l@d@dashtrue
4751   \fi
4752 \fi%
4753 %

4754 \ifl@d@dash%
4755   \ifboolexpr{togl{fulllines@} or test{\ifcsemt{Xendtwolines@}\
@currentseries}}}%
4756   {%
4757   {%
4758     \setistwofollowinglines{#1}{#2}{#4}{#5}%
4759     \ifboolexpr{%
4760       (%
4761         togl {Xendtwolinesbutnotmore@\@currentseries}%
4762         and not%
4763         (%
4764           bool {istwofollowinglines@}%
4765         )%
4766       )%
4767       or%
4768       (%
4769         (not test{\ifnumequal{#1}{#4}})%
4770         and togl{Xendtwolinesonlyinsamepage@\@currentseries}%
4771       )%
4772     }%
4773   }%
4774   {%
4775     \l@d@dashfalse%
4776     \l@d@Xtwolinestrue%
4777     \l@d@elinfalse%
4778     \l@d@eslfalse%
4779     \ifcsemt{Xendmorethantwolines@\@currentseries}%
4780     {%
4781       \ifistwofollowinglines@\else%
4782         \l@d@Xmorethantwolinestrue%
4783       \fi%
4784     }%
4785   }%
4786 }%
4787 \fi%
4788 %

```

End of `\setprintendlines`.

```
4789 }%
4790 %
```

`\printendlines` Now we are ready to print it all.

```
4791 \def\printendlines#1|#2|#3|#4|#5|#6|#7|#8|{%
4792   \begingroup
4793   \setprintendlines{#1}{#2}{#3}{#4}{#5}{#6}%
4794   %
```

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, start the starting line box, if needed.

```
4795   \ifdimequal{\csuse{Xendboxstartlinenum@\@currentseries}}{0pt}%
4796   {\bgroup}%
4797   {\leavevmode\hbox to \csuse{Xendboxstartlinenum@\@currentseries}\bgroup
\hfill}}%
4798   %
```

Then, print the starting page number-

```
4799   \ifboolexpr{%
4800     (%
4801       test{\ifcsstring{prevpagenum@\@currentseries}{#1}}%
4802       and not%
4803       (togl{Xendpagenumberonlyfirsttifsingl@\@currentseries} and bool{
1@d@pnum})}%
4804     )%
4805     or%
4806     (%
4807       test {\ifcsstring{prevpagerange@\@currentseries}{#1-#4}}%
4808     )%
4809   }%
4810   {%
4811     \ifcsempy{Xendsympagenum@\@currentseries}%
4812     {\hspace{\csuse{Xendinplaceofpagenumber@\@currentseries}}}%
4813     {\csuse{Xendsympagenum@\@currentseries}}%
4814   }%
4815   {%
4816     \wrap@edcrossref{\@this@crossref@start}{\printnnum{#1}}%
4817   }%
4818   %
```

Then, determine what must be printed before the start line.

```
4819   \ifl@d@dash%
4820   \ifl@d@pnum%
```

```

4821 \csuse{Xendlineprefixsingle@ \@currentseries}%
4822 \else%
4823 \ifcempty{Xendlineprefixmore@ \@currentseries}%
4824 {\csuse{Xendlineprefixsingle@ \@currentseries}}%
4825 {\csuse{Xendlineprefixmore@ \@currentseries}}%
4826 \fi%
4827 \else%
4828 \csuse{Xendlineprefixsingle@ \@currentseries}%
4829 \fi%
4830 %

```

The print the starting line, followed, if needed, by the side flag and the starting sub line number.

```

4831 \wrap@edcrossref{\@this@crossref@start}{%
4832 \ifledRcol@%
4833 \linenumrepR{#2}%
4834 \else%
4835 \linenumrep{#2}%
4836 \fi%
4837 }%
4838 \iftoggle{Xendlineflag@ \@currentseries}{\ifledRcol@ \@Rlineflag\fi}{}%
4839 \ifl@d@ssub%
4840 \csuse{Xendsublinesep@ \@currentseries}%
4841 \wrap@edcrossref{\@this@crossref@start}{%
4842 \ifledRcol@%
4843 \sublinenumrepR{#3}%
4844 \else%
4845 \sublinenumrep{#3}%
4846 \fi%
4847 }%
4848 \fi%
4849 %

```

Close the box.

```

4850 \egroup%
4851 %

```

Open the box for the ending line number.

```

4852 \ifdimequal{\csuse{Xendboxendlinenum@ \@currentseries}}{0pt}%
4853 {\bgroup}%
4854 {\hbox to \csuse{Xendboxendlinenum@ \@currentseries}\bgroup}%
4855 %

```

Print the dash + the ending line number, or the line number range symbol.

```

4856 \ifl@d@xtwolines%
4857 \ifl@d@Xmorethantwolines%
4858 \csuse{Xendmorethantwolines@ \@currentseries}%
4859 \else%
4860 \csuse{Xendtwolines@ \@currentseries}%

```

```

4861 \fi%
4862 \else%
4863 \ifl@d@dash%
4864 \ifdefined\linerangesep%
4865 \linerangesep%
4866 \else%
4867 \csuse{Xendlinrangeseparator@\@currentseries}%
4868 \fi%
4869 \fi%
4870 %

```

Print the ending page number.

```

4871 \ifl@d@pnum%
4872 \ifcsstring{prevpagerange@\@currentseries}{#1-#4}%
4873 {%
4874 \ifcempty{Xendsympagenum@\@currentseries}%
4875 {\hspace{\csuse{Xendinplaceofpagenumber@\@currentseries}}}%
4876 {\csuse{Xendsympagenum@\@currentseries}}%
4877 }%
4878 {%
4879 \wrap@edcrossref{\@this@crossref@end}\printnpnum{#4}%
4880 }%
4881 \fi%
4882 %

```

Print the ending line number, with, if needed, the line prefix, and followed by the side flag and the subline number.

```

4883 \ifl@d@elin%
4884 \ifl@d@pnum\csuse{Xendlineprefixsingle@\@currentseries}\fi%
4885 \wrap@edcrossref{\@this@crossref@end}{%
4886 \ifledRcol%
4887 \linenumrepR{#5}%
4888 \else%
4889 \linenumrep{#5}%
4890 \fi%
4891 }%
4892 \iftoggle{Xendlineflag@\@currentseries}{\ifledRcol@\@Rlineflag\fi}{}%
4893 \fi%
4894 \ifl@d@esl%
4895 \ifl@d@elin%
4896 \csuse{Xendsublinesep@\@currentseries}%
4897 \fi%
4898 \wrap@edcrossref{\@this@crossref@end}{%
4899 \ifledRcol%
4900 \sublinenumrepR{#6}%
4901 \else%
4902 \sublinenumrep{#6}%
4903 \fi%
4904 }%
4905 \fi%

```



```

4906 \fi%
4907 %

```

Close the ending line box.

```

4908 \ifdimequal{\csuse{Xendboxendlinenum@\@currentseries}}{0pt}%
4909 {}%
4910 {\hfill}%Prevent underfull hbox
4911 \egroup%
4912 %

```

And, finally, save, if needed, the current page number for the Xendpagenumberonlyfirst hooks.

```

4913 \iftoggle{Xendpagenumberonlyfirst@\@currentseries}%
4914 {\iftoggle{Xendpagenumberonlyfirstintwo@\@currentseries}%
4915 {\csxdef{prevpagerange@\@currentseries}{#1-#4}}%
4916 {\csxdef{prevpagenum@\@currentseries}{#4}}%
4917 }%
4918 {}%
4919 %

```

Now, the end of \printendlines macro.

```

4920 \endgroup%
4921 }%
4922
4923 %

```

**\printnpnum** A macro to print a page number in an endnote. Should not be override anymore

```

4924 \newcommand*{\printnpnum}[1]{\csuse{Xendbeforepagenumber@\@currentseries}
4925 #1\csuse{Xendafterpagenumber@\@currentseries}}
4926 %

```

## XX Generate series of notes

In this section, X means the name of the series (A, B etc.)

**\series** \series\series creates one more new series. It is a public command, which just loops on the private command \newseries@.

```

4927 \newcommand{\newseries}[1]{%
4928 \def\do##1{\newseries@{##1}}%
4929 \docsvlist{#1}
4930 }
4931 %

```

**\@series** The \series@ macro is an etoolbox list, which contains the name of all series.

```

4932 \newcommand{\@series}{}
4933 %

```

The command `\newseries@series` creates a new series of the footnote.

```

\newseries@34 \newcommand{\newseries@}[1]{
4935 %

```

### XX.1 Test if series is still existing

```

4936 \xifinlist{#1}{\@series}{\led@warn@SeriesStillExist{#1}}%
4937 {%
4938 %

```

### XX.2 Init specific to reledpar

When calling `\newseries@` after having loaded `reledpar`, we need to load specific setting.

```

4939 \ifdefined\newseries@par%
4940 \newseries@par{#1}%
4941 \fi%
4942 %

```

### XX.3 For critical footnotes

Critical footnotes are those which start with letters. We look for the `\nocritical` option of `reledmac`.

```

4943 \unless\ifnocritical@
4944 %

```

#### XX.3.1 Options

```

4945 \newtoggle{Xlineflag@#1}
4946 \newtoggle{Xparindent@#1}
4947 \newtoggle{Xlemmadisablefontselection@#1}
4948 \csgdef{Xwrapcontent@#1}{}%
4949 \csgdef{Xbeforeinserting@#1}{}%
4950 \csgdef{Xhangindent@#1}{Opt}%
4951 \csgdef{Xragged@#1}{}%
4952 \csgdef{Xhsizetwocol@#1}{0.45 \hsize}%
4953 \csgdef{Xhsizethreecol@#1}{.3 \hsize}%
4954 \csgdef{Xcolalign@#1}{\raggedright}%
4955 \csgdef{Xnotenumfont@#1}{\normalfont}%
4956 \csgdef{Xnotefontsize@#1}{\footnotesize}%
4957 \csgdef{Xbhooknote@#1}{}%

```

```

4958 \csgdef{Xbhookgroup@#1}{}%
4959
4960 \csgdef{Xboxlinenum@#1}{Opt}%
4961 \csgdef{Xboxlinenumalign@#1}{L}%
4962
4963 \csgdef{Xboxstartlinenum@#1}{Opt}%
4964 \csgdef{Xboxendlinenum@#1}{Opt}%
4965
4966 \csgdef{Xboxsymlinenum@#1}{Opt}%
4967 \newtoggle{Xnumberonlyfirstinline@#1}%
4968 \newtoggle{Xgroupbyline@#1}%
4969 \newtoggle{Xgroupbylineseparetwolines@#1}%
4970 \newtoggle{Xnumberonlyfirstintwolines@#1}%
4971 \csgdef{Xtwolines@#1}{}%
4972 \csgdef{Xmorethantwolines@#1}{}%
4973 \csgdef{Xsublinesep@#1}{\fullstop}%
4974 \csgdef{Xpagelinesep@#1}{\csname Xsublinesep@#1\endcsname}%for
backward compatibility, call Xsublinesep@#1
4975 \newtoggle{Xtwolinesbutnotmore@#1}%
4976 \newtoggle{Xtwolinesonlyinsamepage@#1}%
4977 \newtoggle{Xonlypstart@#1}%
4978 \newtoggle{Xpstarteverytime@#1}%
4979 \newtoggle{Xpstart@#1}%
4980 \newtoggle{Xstanza@#1}%
4981 \csgdef{Xstanzaseparator@#1}{}%
4982 \csgdef{Xsymlinenum@#1}{}%
4983 \newtoggle{Xnonumber@#1}%
4984 \csgdef{Xbeforenumber@#1}{Opt}%
4985 \csgdef{Xtxtbeforenumber@#1}{}%
4986 \csgdef{Xafternumber@#1}{0.5em}%
4987 \newtoggle{Xnonbreakableafternumber@#1}%
4988 \csgdef{Xbeforesymlinenum@#1}{\csuse{Xbeforenumber@#1}}%
4989 \csgdef{Xaftersymlinenum@#1}{\csuse{Xafternumber@#1}}%
4990 \csgdef{Xinplaceofnumber@#1}{1em}%
4991 \global\cslet{Xlemmaseparator@#1}{\rbracket}%
4992 \csgdef{Xbeforelemmaseparator@#1}{0em}%
4993 \csgdef{Xafterlemmaseparator@#1}{0.5em}%
4994 \csgdef{Xinplaceoflemmaseparator@#1}{1em}%
4995 \csgdef{Xbeforenotes@#1}{1.2em \@plus .6em \@minus .6em}%
4996 \csgdef{Xafterrule@#1}{Opt}%
4997
4998 \csgdef{Xtxtbeforenotes@#1}{%
4999 \newtoggle{Xtxtbeforenotes@#1@typeset}%Not directly used by user,
but internal
5000
5001 \csgdef{Xmaxhnotes@#1}{0.8\vsizel}
5002 \newtoggle{Xnoteswidthliketwocolumns@#1}%
5003 \csgdef{Xparafootsep@#1}{}%
5004 \csgdef{Xafternote@#1}{1em plus.4em minus.4em}%
5005 \csgdef{Xlinrangeseparator@#1}{\endashchar}%

```

```

5006 \csgdef{Xlemmafont@#1}{}%
5007 \csgdef{Xwraplemma@#1}{%
5008 \csgdef{Xwidth@#1}{\hspace}%
5009 %
5010 %

```

### XX.3.2 Create inserts, needed to add notes in foot

As regards inserts, see chapter 15 of *The TeXbook* by D. Knuth.

```

5011 \expandafter\newinsert\csname #1footins\endcsname%
5012 \unless\ifnoledgroup%
5013 \expandafter\newinsert\csname mp#1footins\endcsname%
5014 \fi%
5015 %

```

### 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.

```

5016 \global\newbool{parapparatus@}\expandafter\newcommand\expandafter
5017 *}{\expandafter\newcommand}\csname #1footnote\endcsname[2] []{%
5018 \if@edtext@secondarg%
5019 \ifledRcol%
5020 \ifcsstring{Xonlyside@#1}{L}{\
5021 led@error@note@called@onrightside{#1footnote}}}%
5022 \else%
5023 \ifcsstring{Xonlyside@#1}{R}{\
5024 led@error@note@called@onleftside{#1footnote}}}%
5025 \fi%
5026 \beginngroup%
5027 \newcommand{\content}{##2}%
5028 \ifnumberedpar%
5029 \ifledRcol%
5030 \ifluatex%
5031 \footnotelang@lua[R]%
5032 \fi%
5033 \@ifundefined{xpg@main@language}%if polyglossia
5034 {}%
5035 {\footnotelang@poly[R]}%
5036 \footnoteoptions@{R}{#1}{true}%
5037 \xright@appenditem{%
5038 \ifbool{indtl@innote}%
5039 {\unexpanded{\let\index\nindex}}%
5040 {}%
5041 \ifbool{indtl@notenumber}%
5042 {\unexpanded{\let\index\nindex}}%There is no note
5043 ...number so
5044 {}%
5045 \noexpand\Xnote@true%

```

```

5042         \noexpand\prepare@Xprenotes{#1}%
5043         \noexpand\prepare@edindex@fornote{\l@d@nums}%
5044         \unexpanded{\def\sw@list@inedtext}{\expandafter\
unexpanded\expandafter{\sw@inthisedtext}}}%The value of the \sw@inthisedtext
of current \edtext will be pushed to \sw@list@inedtext when the notes are
expanded.
5045         \noexpand\setcounter{stanzaR}{\the\c@stanzaR}%Save
stanzaR counter for footnote
5046         \unexpanded{\def\@this@crossref@start}{\theedtext:
start}%
5047         \unexpanded{\def\@this@crossref@end}{\theedtext:end}%
5048         \expandonce{\@beforeinsertofthisedtext}% Internal for
now, no reason to make it public
5049         \noexpand\csuse{v#1footnote}{#1}%
5050         {\l@d@nums}{\expandonce\@tag}{\expandonce\content}}
%
5051         \noexpand\Xnote@false%
5052         \unexpanded{\advance\@edindex@fornote@m@ne}%
5053         \unexpanded{\undef\@this@crossref@start}%
5054         \unexpanded{\undef\@this@crossref@end}%
5055         \ifbool{indtl@innote}%
5056         {\unexpanded{\let\index\orig@@index}}%
5057         {}%
5058         \ifbool{indtl@notenumber}%
5059         {\unexpanded{\let\index\orig@@index}}%
5060         {}%
5061         }\to\inserts@listR
5062         \footnoteoptions@{R}{##1}{false}%
5063         \global\advance\insert@countR \@ne%
5064     \else%
5065         \ifluatex%
5066         \footnotelang@lua%
5067         \fi%
5068         \@ifundefined{xpg@main@language}%if polyglossia
5069         {}%
5070         {\footnotelang@poly}%
5071         \footnoteoptions@{L}{##1}{true}%
5072         \xright@appenditem{%
5073         \ifbool{indtl@innote}%
5074         {\unexpanded{\let\index\nindex}}%
5075         {}%
5076         \ifbool{indtl@notenumber}%
5077         {\unexpanded{\let\index\nindex}}}%There is no note
...number so
5078         {}%
5079         \noexpand\Xnote@true%
5080         \noexpand\prepare@Xprenotes{#1}%
5081         \noexpand\prepare@edindex@fornote{\l@d@nums}%
5082         \unexpanded{\def\sw@list@inedtext}{\expandafter\
unexpanded\expandafter{\sw@inthisedtext}}}%The value of the \sw@inthisedtext

```

```

of current edtext will be pushed to \sw@list@inedtext when the notes are
expanded.
5083         \ifl@dpairing%
5084         \noexpand\setcounter{stanzaL}{\the\c@stanzaL}%Save
stanzaR counter for footnote
5085         \fi%
5086         \unexpanded{\def\@this@crossref@start}{\theedtext:
start}%
5087         \unexpanded{\def\@this@crossref@end}{\theedtext:end}%
5088         \expandonce{\@beforeinsertofthisedtext}%Internal for
now, no reason to make it public
5089         \noexpand\csuse{v#1footnote}%
5090         {#1}%
5091         {\l@d@nums}{\expandonce\@tag}{\expandonce\content
}}%
5092         \unexpanded{\undef\@this@crossref@start}%
5093         \unexpanded{\undef\@this@crossref@end}%
5094         \noexpand\Xnote@false%
5095         \unexpanded{\advance\@edindex@fornote@m@one}%
5096         \ifbool{indtl@innote}%
5097         {\unexpanded{\let\index\orig@@index}}%
5098         {}%
5099         \ifbool{indtl@notenumber}%
5100         {\unexpanded{\let\index\orig@@index}}%
5101         {}%
5102         }\to\inserts@list
5103         \global\advance\insert@count \@ne%
5104         \footnoteoptions@{L}{#1}{false}%
5105         \fi
5106         \else
5107         \csuse{v#1footnote}{#1}{\{0|0|0|0|0|0|0\}}{#1}}%
5108         \fi%
5109         \endgroup%
5110         \else%
5111         \led@err@FootnoteNotInSecondArgEdtext{#1}%
5112         \fi%
5113         \ignorespaces%
5114         }
5115 %

```

Create counter used to determine on which page the previous note was called.

```

5116         \expandafter\newcount\csname #1prevpage@num\endcsname%
5117         \expandafter\newcount\csname #1prevpage@numR\endcsname%
5118 %

```

We need to be able to modify reledmac's footnote macros and restore their

```

5119         \global\csletcs{#1@@footnote}{#1footnote}
5120 %

```

**XX.3.4 Set standard display**

```

5121 \Xarrangement@normal{#1}%
5122 %

```

End of for critical footnotes.

```

5123 \fi
5124 %

```

**XX.4 For familiar footnotes**

Familiar footnotes are those which end with letters. We look for the `nofamiliar` option of `reledmac`.

```

5125 \unless\ifnofamiliar@
5126 %

```

**XX.4.1 Options**

```

5127 \newtoggle{parindentX@#1}
5128 \csgdef{wrapcontentX@#1}{}%
5129 \csgdef{hangindentX@#1}{Opt}%
5130 \csgdef{beforeinsertingX@#1}{}%
5131 \csgdef{raggedX@#1}{}%
5132 \csgdef{hsizetwocolX@#1}{0.45 \hsize}%
5133 \csgdef{hsizethreecolX@#1}{.3 \hsize}%
5134 \csgdef{colalignX@#1}{\raggedright}%
5135 \csgdef{notenumfontX@#1}{\normalfont}%
5136 \csgdef{notefontsizeX@#1}{\footnotesize}%
5137 \csgdef{bhooknoteX@#1}{}%
5138 \csgdef{bhookgroupX@#1}{}%
5139 \csgdef{afterruleX@#1}{Opt}
5140 \csgdef{beforenotesX@#1}{1.2em \@plus .6em \@minus .6em}
5141 \csgdef{maxhnotesX@#1}{0.8\vsize}%
5142 \newtoggle{noteswidthliketwocolumnsX@#1}%
5143 \csgdef{parafootsepX@#1}{}%
5144 \csgdef{afternoteX@#1}{1em plus.4em minus.4em}
5145 \csgdef{widthX@#1}{\hsize}%
5146 % End of for familiar footnotes.
5147 % \subsubsection{Create inserts, needed to add notes in foot}
5148 % As regards inserts, see chapter 15 of the TeXBook by D. Knuth.
5149 % \begin{macrocode}
5150 \expandafter\newinsert\csname footins#1\endcsname%
5151 \unless\ifnoledgroup@%
5152 \expandafter\newinsert\csname mpfootins#1\endcsname%
5153 \fi%
5154 %

```

**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.

```

5155 \global\expandafter\newcommand\csname footnote#1\endcsname[1]{%
5156 \begingroup%
5157 \prepare@prenotesX{#1}%
5158 \newcommand{\content}{##1}%
5159 %
5160 %

```

If we use the \csquotes package, we reset quotation level.

```

5161 \ifdefined\csq@qllevel%
5162 \csq@qllevel=0\relax%
5163 \fi%
5164 %

```

If we are preparing parallel typesetting, we cannot just increase the footnote counter. Read reledpar's handbook about that (V.1.2 p. 50).

```

5165 \global\expandafter\advance\csname footnote#1@reading\
endcsname by \@ne%
5166 \ifboolexpr{bool{!@dpairing} or bool{!@dprintingpages} or
bool{!@dprintingcolumns}}{%
5167 \ifcsdef{footnote#1@reading\the\csname footnote#1@reading\
endcsname=typeset}%
5168 {\setcounter{footnote#1}{\csuse{footnote#1@reading\the\
csname footnote#1@reading\endcsname=typeset}}}%
5169 {\setcounter{footnote#1}{\the\csname footnote#1@reading
\endcsname}}}%
5170 }{%
5171 \stepcounter{footnote#1}%
5172 }%
5173 %

```

We also have to check consistency with \onlysideX setting.

```

5174 \ifledRcol%
5175 \ifcsstring{onlysideX@#1}{L}{\
led@error@note@called@onrightside{footnote#1}}}%
5176 \else%
5177 \ifcsstring{onlysideX@#1}{R}{\
led@error@note@called@onleftside{footnote#1}}}%
5178 \fi%
5179 %

```

And now, the feature not depending of whether we are preparing parallel typesetting

```

5180 \protected@csxdef{@thefnmark#1}{\csuse{thefootnote#1}}%
5181 \nottoggle{nomk@}%Nomk is set to true when using \
footnoteXnomk with \parpackage
5182 {\csuse{@footnotemark#1}}%

```



```

5183         {}%
5184         \ifluatex%
5185             \xdef\footnote@luatextextdir{\the\textdir}%
5186             \xdef\footnote@luatexpardir{\the\pardir}%
5187         \fi%
5188         \if@ledgroup%
5189             \led@set@index@fornote{#1}%
5190         \fi%
5191         \csuse{vfootnote#1}{#1}{\expandonce\content}\m@mmf@prepare%
5192         \ifbool{indtl@innote}%
5193             {\let\index\orig@@index}%
5194             {}%
5195         \ifbool{indtl@notenumber}%
5196             {\let\index\orig@@index}%
5197             {}%
5198     \endgroup%
5199 }
5200 %

```

Then define the counters. The  $\TeX$  counter `footnoteX` is the only one manipulated by the user. This is the one which is printed. The  $\TeX$  counter `\footnoteX@reading` is increased at each footnote. It is used for hyperlinks, for using `hyperlink` package, and for getting the correct footnote number when using parallel typesetting (V.1.2 p. 50).

```

5201     \newcounter{footnote#1}
5202     \global\expandafter\renewcommand\csname thefootnote#1\endcsname{\
arabic{footnote#1}}
5203     \expandafter\newcount\csname footnote#1@reading\endcsname%
5204 %

```

Create counter used to determine on which page the previous note was called.

```

5205     \expandafter\newcount\csname prevpage#1@num\endcsname%
5206     \expandafter\newcount\csname prevpage#1@numR\endcsname%
5207 %

```

Add `\let\footnoteX\@gobble` to `\no@expands`.

```

5208     \expandafter\gappto\expandafter\no@expands\expandafter{\expandafter\
let\csname footnote#1\endcsname\@gobble}%
5209 %

```

And now, define `\footnoteXmark` and `\footnoteXtext`, equivalent to classical `\footnotemark` and `\footnotetext`.

```

5210     \expandafter\newcommand\csname footnote#1mark\endcsname{%
5211         \begingroup%
5212         \prepare@prenotesX{#1}%
5213         \stepcounter{footnote#1}%
5214         \protected@csxdef{@thefnmark#1}{\csuse{thefootnote#1}}%
5215         \csuse{@footnotemark#1}%
5216         \m@mmf@prepare%
5217     \endgroup%

```

```

5218 }%
5219 \expandafter\newcommand\csname footnote#1text\endcsname[1]{%
5220   \begingroup%
5221     \csuse{vfootnote#1}{#1}{\expandonce{##1}}%
5222   \endgroup%
5223 }%
5224 %

```

Do not forget to initialize the series.

```

5225 \arrangementX@normal{#1}%
5226 \fi
5227 %

```

## XX.5 The endnotes

Endnotes are commands like `\Xendnote`, where `X` is a series letter. First, we check for the `noend` options.

```

5228 \unless\ifnoend@
5229 %

```

### XX.5.1 The auxiliary file

Endnotes of all varieties are saved up in a file, one by series, typically named `<jobname>.Xend`. `\l@d@end` is the output stream number for this file, and `\ifl@dend@X` is a flag that is true when the file is open.

```

5230 \expandafter\newwrite\csname l@d@#1end\endcsname%
5231 \expandafter\newif\csname ifl@dend@#1\endcsname%
5232 %

```

### 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.

```

5233 \global\expandafter\newcommandx\csname #1endnote\endcsname[2][1,
5234 usedefault]{%
5235   \bgroup%
5236   \newlinechar='40%
5237   \global\@noneed@Footnotetrue%
5238   \newcommand{\content}{##2}%
5239   \stepcounter{labidx}%
5240   \expandafter\immediate\expandafter\write\csname l@d@#1end\endcsname{%

```

```

5241 \unexpanded{\def\sw@list@inedtext}{\expandafter\unexpanded\
expandafter{\sw@inthisedtext}}\@percentchar\space%Explicit space, to add a
linebreak in the output file
5242 \expandafter\string\csname #1end\endcsname%
5243 {\ifnumberedpar@\l@d@nums\fi}%
5244 {\ifnumberedpar@\expandonce\@tag\fi}%
5245 {\expandonce\content}%
5246 {\#1}%
5247 {\unexpanded{##1}}%
5248 {\ifl@edRcol R\else L\fi}%
5249 {\theedtext}%
5250 \@percentchar%
5251 }%
5252 \egroup%
5253 \ignorespaces%
5254 }%
5255 %

```

### XX.5.3 Tools

The `\Xtoendnotes` command inserts any arbitrary content into the endnote file. It is an alias of the more generalist `\addtoendnotes`

```

5256 \global\expandafter\newcommand\csname #1toendnotes\endcsname[1]{%
5257 \toendnotes[#1]{##1}%
5258 }%
5259 %
5260 \expandafter\WithSuffix\expandafter\newcommand\csname #1toendnotes\
5261 endcsname*[1]{%
5262 \toendnotes*[#1]{##1}%
5263 }%
5264 %
5265 %

```

### XX.5.4 Internal commands

`\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 `\@gobbleseven`, which just skips over its seven arguments.

```

5266 \global\cslet{#1end}{\@gobbleseven}
5267 %
5268 %

```

We need to store the number of times `\doendnotesbysection` is called for one series.

```

5269 \global\expandafter\newcount\csname #1end@bysection\endcsname%
5270 %

```

### XX.5.5 The options

```

5271 \csgdef{Xendwraplemma@#1}{%
5272 \csgdef{Xendwrapcontent@#1}{}%
5273 \csgdef{Xendtwolines@#1}{}%
5274 \csgdef{Xendmoreethantwolines@#1}{}%
5275 \newtoggle{Xendtwolinesbutnotmore@#1}{}%
5276 \newtoggle{Xendtwolinesonlyinsamepage@#1}{}%
5277 \newtoggle{Xendlemmadisablefontselection@#1}{}%
5278 \csgdef{Xendnotenumfont@#1}{\normalfont}%
5279 \csgdef{Xendnotefontsize@#1}{\footnotesize}%
5280 \csgdef{Xendbhooknote@#1}{}%
5281
5282 \csgdef{Xendsublinesep@#1}{\fullstop}%
5283
5284 \csgdef{Xendbeforenumber@#1}{Opt}
5285 \csgdef{Xendafternumber@#1}{0.5em}
5286
5287 \csgdef{Xendboxlinenum@#1}{Opt}%
5288 \csgdef{Xendboxlinenumalign@#1}{L}%
5289
5290 \csgdef{Xendboxstartlinenum@#1}{Opt}%
5291 \csgdef{Xendboxendlinenum@#1}{Opt}%
5292
5293 \csgdef{Xendlemmaseparator@#1}{}%
5294 \csgdef{Xendbeforelemmaseparator@#1}{Oem}%
5295 \csgdef{Xendafterlemmaseparator@#1}{0.5em}%
5296 \csgdef{Xendinplaceoflemmaseparator@#1}{0.5em}%
5297
5298 \newtoggle{Xendparagraph@#1}%
5299 \csgdef{Xendafternote@#1}{1em plus.4em minus.4em}%
5300 \csgdef{Xendsep@#1}{}%
5301
5302 \csgdef{Xendinplaceofnumber@#1}{Opt}%
5303 \newtoggle{Xendnonumber@#1}%
5304
5305 \csgdef{Xendhangindent@#1}{Opt}%
5306 \newtoggle{Xendnumberonlyfirstinline@#1}%
5307 \newtoggle{Xendnumberonlyfirstintwolines@#1}%
5308
5309 \csgdef{Xendbeforesymlinenum@#1}{\csuse{Xendbeforenumber@#1}}%
5310 \csgdef{Xendaftersymlinenum@#1}{\csuse{Xendafternumber@#1}}%
5311 \csgdef{Xendsymlinenum@#1}{}%
5312 \csgdef{Xendboxsymlinenum@#1}{Opt}%
5313
5314 \csgdef{Xendbhooklinenumber@#1}{}%
5315 \csgdef{Xendehooklinenumber@#1}{}%
5316 \csgdef{Xendbhookinplaceofnumber@#1}{}%
5317 \csgdef{Xendehookinplaceofnumber@#1}{}%
5318

```

```

5319 \csgdef{Xendlinrangeseparator@#1}{\endashchar}%
5320
5321 \csgdef{Xendbeforepagenumber@#1}{p.}%
5322 \csgdef{Xendafterpagenumber@#1}{) }%
5323 \csgdef{Xendlineprefixsingle@#1}{}%
5324 \csgdef{Xendlineprefixmore@#1}{}%
5325
5326 \newtoggle{Xendlineflag@#1}
5327
5328 \csgdef{Xendlemmafont@#1}{}%
5329
5330 \newtoggle{Xendpagenumberonlyfirst@#1}%
5331 \newtoggle{Xendpagenumberonlyfirsttifsingl@#1}%
5332 \newtoggle{Xendpagenumberonlyfirstintwo@#1}%
5333 \csgdef{Xendsympagenum@#1}{}%
5334 \csgdef{Xendinplaceofpagenumber@#1}{0pt}%
5335
5336 %

```

End of endnotes declaration

```

5337 \fi%
5338 %

```

Dump series in \@series

```

5339 \listxadd{\@series}{#1}
5340 }
5341 }% End of \newseries
5342 %

```

## XX.6 Init standards series (A,B,C,D,E)

```

5343 \expandafter\newseries\expandafter{\default@series}
5344 %

```

## 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.

```

5345 \newcommand{\seriesatbegin}[1]{%
5346   \StrDel{\@series}{#1}[\@series]%
5347   \edef\@new{%
5348     \listxadd{\@new}{#1}%
5349     \listxadd{\@new}{\@series}%
5350     \xdef\@series{\@new}%
5351   }
5352 %

```

`\seriesatend` And `\seriesatend` moves the series to the end of the list.

```

5353 \newcommand{\seriesatend}[1]{%
5354   \StrDel{\@series}{#1}[\@series]%
5355   \edef\@new{%
5356     \listadd{\@new}{\@series}%
5357     \listadd{\@new}{#1}%
5358     \xdef\@series{\@new}%
5359   }
5360   %

```

## XXI.2 Test series order

`\ifseriesbefore` `\ifseriesbefore{<seriesA>}{<seriesB>}{<true>}{<false>}` expands `<true>` if `<seriesA>` is printed before `<seriesB>`, expands `<false>` otherwise.

```

5361 \newcommand{\ifseriesbefore}[4]{%
5362   \StrPosition{\@series}{#1}[\@first]%
5363   \StrPosition{\@series}{#2}[\@second]%
5364   \ifnumgreater{\@second}{\@first}{#3}{#4}%
5365 }
5366 %

```

### XXI.2.1 Get the first series

In some specific case, we need to know the first series of the list of series.

```

\@getfirstseries67 \newcommand{\@getfirstseries}{%
5368   \ifdefempty{\@series}%
5369   {\xdef\@firstseries{}}%
5370   {\StrChar{\@series}{1}[\@firstseries]}%
5371 }%
5372 %

```

## 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`.

```

5373 \newcommandx{\settoggle@series}[5][4,5,usedefault]{%
5374   \def\do##1{%
5375     \global\settoggle{#2@##1}{#3}%
5376     \ifstrequal{#4}{critical}{
5377       \csuse{Xarrangement@}\csuse{series@display##1}}{##1}%
5378     }{}
5379     \ifstrequal{#4}{familiar}{
5380       \csuse{arrangementX@}\csuse{series@displayX##1}}{##1}%
5381     }{}
5382   }%
5383   \ifstreempty{#1}{%
5384     \dolistloop{\@series}%
5385     \ifstreempty{#5}{}{%
5386       \docsvlist{#5}%
5387     }
5388   }%
5389   {%
5390     \docsvlist{#1}%
5391   }%
5392 }
5393 %

```

`\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`.

```

5394 \newcommandx{\setcommand@series}[5][4,5,usedefault]{%
5395   \def\do##1{
5396     \csgdef{#2@##1}{#3}
5397     \ifstrequal{#4}{critical}{%
5398       \csuse{Xarrangement@}\csuse{series@display##1}}{##1}%
5399     }{}
5400     \ifstrequal{#4}{familiar}{%
5401       \csuse{arrangementX@}\csuse{series@displayX##1}}{##1}%
5402     }{}%
5403   }%
5404   \ifstreempty{#1}{%
5405     \dolistloop{\@series}%
5406     \ifstreempty{#5}{}{%
5407       \docsvlist{#5}
5408     }
5409   }%
5410   {%
5411     \docsvlist{#1}%
5412   }%
5413 }%
5414 %

```

### 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`.

```

5415 \newcommandx{\newhookcommand@series}[2][2,usedefault]{%
5416   \global\expandafter\newcommand\expandafter*\csname #1\endcsname[2][]{%
5417     \setcommand@series{##1}{#1}{##2}[][#2]%
5418   }%
5419   \ifstreempty{#2}{}{%
5420     \def\do##1{%

```



```

5421 \global\expandafter\newcommand\expandafter*\csname #1##1\endcsname
5422 [1]{%
5423     \csuse{#1}[##1]{####1}%
5424     }%
5425     \docsvlist{#2}%
5426     }%
5427 }
5428 %

```

**\newhooktoggle@series** `\newhooktoggle@series`\command names is a generic command to add new commands for a new toggle hook, like `\Xnumberonlyfirstinline`. The second argument is also used to create commands named `\<hookname><pseudoserries>`, like `\Xtwolinesbutnotmoreappref`.

```

5429 \newcommandx{\newhooktoggle@series}[2][2,usedefault]{%
5430 \global\expandafter\newcommandx\expandafter*\csname #1\endcsname[2][1,2={
5431 true},usedefault]{%
5432     \settoggle@series{##1}{#1}{##2}[][#2]%
5433     }%
5434     \ifstrepty{#2}{-}{%
5435         \def\do##1{%
5436             \global\expandafter\newcommand\expandafter*\csname #1##1\endcsname{%
5437                 \csuse{#1}[##1]%
5438             }%
5439             \docsvlist{#2}%
5440         }%
5441     }
5442 %

```

**\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, depending of type os notes

```

5443 \newcommand{\newhooktoggle@series@reload}[2]{%
5444 \global\expandafter\newcommandx\expandafter*\csname #1\endcsname[2][1,2={
5445 true},usedefault]{%
5446     \settoggle@series{##1}{#1}{##2}[#2]%
5447     }%
5448 %

```

**\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.

```

5449 \newcommand{\newhookcommand@series@reload}[2]{%
5450 \global\expandafter\newcommand\expandafter*\csname #1\endcsname[2][]{%
5451     \setcommand@series{##1}{#1}{##2}[#2]%
5452     }%

```

```

5453 }
5454 %

```

### 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.

```

5455 \unless\ifnocritical@
5456 \newhookcommand@series{Xwrapcontent}%
5457 \newhookcommand@series{Xbeforeinserting}%
5458 \newhookcommand@series{Xlemmafont}%
5459 \newhookcommand@series{Xwraplemma}%
5460 \newhooktoggle@series{Xparindent}
5461 \newhookcommand@series{Xhangindent}
5462 \newhookcommand@series{Xragged}
5463 \newhookcommand@series{Xhsizetwocol}
5464 \newhookcommand@series{Xhsizethreecol}
5465 \newhookcommand@series{Xcolalign}%
5466 \newhookcommand@series{Xnotenumfont}
5467 \newhookcommand@series{Xbhooknote}
5468 \newhookcommand@series@reload{Xbhookgroup}{critical}
5469 \newhookcommand@series{Xboxsymlinenum}%
5470 \newhookcommand@series{Xsymlinenum}
5471 \newhookcommand@series{Xbeforenumber}
5472 \newhookcommand@series{Xtxtbeforenumber}
5473 \newhookcommand@series{Xafternumber}
5474 \newhookcommand@series{Xbeforesymlinenum}
5475 \newhookcommand@series{Xaftersymlinenum}
5476 \newhookcommand@series{Xinplaceofnumber}
5477 \newhookcommand@series{Xlemmaseparator}
5478 \newhookcommand@series{Xbeforelemmaseparator}
5479 \newhookcommand@series{Xafterlemmaseparator}
5480 \newhookcommand@series{Xinplaceoflemmaseparator}
5481 \newhookcommand@series{Xtxtbeforenotes}
5482 \newhookcommand@series@reload{Xafterrule}{critical}
5483 \newhooktoggle@series{Xnumberonlyfirstinline}
5484 \newhooktoggle@series{Xnumberonlyfirstintwolines}
5485 \newhooktoggle@series{Xgroupbyline}%
5486 \newhooktoggle@series{Xgroupbylineseparetwolines}%
5487 \newhooktoggle@series{Xnonumber}
5488 \newhooktoggle@series{Xpstart}
5489 \newhooktoggle@series{Xpstarteverytime}%
5490
5491 \newhooktoggle@series{Xstanza}%
5492 \newhookcommand@series{Xstanzaseparator}%
5493
5494 \newhooktoggle@series{Xonlypstart}
5495 \newhooktoggle@series{Xnonbreakableafternumber}

```

```

5496 \newhooktoggle@series{Xlemmadisablefontselection}
5497 \newhookcommand@series@reload{Xmaxhnotes}{critical}
5498 \newhookcommand@series@reload{Xbeforenotes}{critical}
5499 \newhooktoggle@series@reload{Xnoteswidthliketwocolumns}{critical}%
5500 \newhookcommand@series@reload{Xnotefontsize}{critical}
5501
5502 \newhookcommand@series{Xboxlinenum}%
5503 \newhookcommand@series{Xboxlinenumalign}%
5504
5505 \newhookcommand@series{Xboxstartlinenum}%
5506 \newhookcommand@series{Xboxendlinenum}%
5507
5508 \newhookcommand@series{Xafternote}%
5509 \newhookcommand@series{Xparafootsep}
5510
5511 \newhookcommand@series@reload{Xwidth}{critical}%
5512
5513 \ifundef{\Xhsize}%
5514 {%
5515   \newcommandx{\Xhsize}[2][1,usedefault]{%
5516     \led@warning@Xhsize@deprecated%
5517     \Xwidth[#1]{#2}%
5518   }%
5519 }%
5520 {}%
5521 \fi
5522 \newhooktoggle@series{Xlineflag}[appref,SEref]
5523 \newhookcommand@series{Xtwolines}[appref,SEref]
5524 \newhookcommand@series{Xmorethantwolines}[appref,SEref]
5525 \newhookcommand@series{Xsublinesep}[appref,SEref,side]%
5526 \newhookcommand@series{Xpagelinesep}[appref,SEref,side]%
5527 \newhooktoggle@series{Xtwolinesbutnotmore}[appref,SEref]
5528 \newhooktoggle@series{Xtwolinesonlyinsamepage}[appref,SEref]
5529 \newhookcommand@series{Xlinrangeseparator}[appref,SEref]
5530 %

```

### XXI.3.5 Options for familiar notes

Before generating the optional commands for familiar notes, we check the `\nofamiliar` option.

```

5531 \unless\ifnofamiliar@
5532   \newhookcommand@series{wrapcontentX}%
5533   \newhookcommand@series{beforeinsertingX}%
5534   \newhooktoggle@series{parindentX}
5535   \newhookcommand@series{hangindentX}
5536   \newhookcommand@series{raggedX}
5537   \newhookcommand@series{hsizetwocolX}
5538   \newhookcommand@series{hsizethreecolX}
5539   \newhookcommand@series{colalignX}%

```

```

5540 \newhookcommand@series{notenumfontX}
5541 \newhookcommand@series{bhooknoteX}
5542 \newhookcommand@series@reload{bhookgroupX}{familiar}
5543 \newhookcommand@series@reload{beforenotesX}{familiar}
5544 \newhookcommand@series@reload{maxhnotesX}{familiar}
5545 \newhooktoggle@series@reload{noteswidthliketwocolumnsX}{familiar}%
5546 \newhookcommand@series@reload{afterruleX}{familiar}
5547 \newhookcommand@series@reload{notefontsizeX}{familiar}
5548 \newhookcommand@series{afternoteX}
5549 \newhookcommand@series{parafootsepX}
5550 \newhookcommand@series@reload{widthX}{familiar}%
5551 \ifundef{\hsizeX}%
5552 {%
5553   \newcommandx{\hsizeX}[2][1,usedefault]{%
5554     \led@warning@hsizeX@deprecated%
5555     \widthX[#1]{#2}%
5556   }%
5557 }%
5558 {}%
5559 \fi
5560 %

```

### 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.

```

5561 \unless\ifnoend@
5562   \newhookcommand@series{Xendwraplemma}
5563   \newhookcommand@series{Xendwrapcontent}
5564   \newhookcommand@series{Xendnotenumfont}
5565   \newhookcommand@series{Xendlemmafont}%
5566   \newhookcommand@series{Xendbhooknote}
5567
5568   \newhookcommand@series{Xendboxlinenum}%
5569   \newhookcommand@series{Xendboxlinenumalign}%
5570
5571   \newhookcommand@series{Xendboxstartlinenum}%
5572   \newhookcommand@series{Xendboxendlinenum}%
5573
5574   \newhookcommand@series{Xendnotefontsize}
5575   \newhooktoggle@series{Xendlemmadisablefontselection}
5576   \newhookcommand@series{Xendlemmaseparator}
5577   \newhookcommand@series{Xendbeforelemmaseparator}
5578   \newhookcommand@series{Xendafterlemmaseparator}
5579   \newhookcommand@series{Xendinplaceoflemmaseparator}
5580
5581   \newhookcommand@series{Xendbeforenumber}%
5582   \newhookcommand@series{Xendafternumber}%
5583

```

```

5584 \newhooktoggle@series{Xendparagraph}
5585 \newhookcommand@series{Xendafternote}
5586 \newhookcommand@series{Xendsep}
5587
5588 \newhookcommand@series{Xendinplaceofnumber}%
5589 \newhooktoggle@series{Xendnonumber}%
5590
5591 \newhooktoggle@series{Xendnumberonlyfirstinline}%
5592 \newhooktoggle@series{Xendnumberonlyfirstintwolines}%
5593
5594 \newhookcommand@series{Xendsymmlinenumber}%
5595 \newhookcommand@series{Xendbeforesymmlinenumber}%
5596 \newhookcommand@series{Xendaftersymmlinenumber}%
5597 \newhookcommand@series{Xendboxsymmlinenumber}%
5598
5599 \newhookcommand@series{Xendbhooklinenumber}%
5600 \newhookcommand@series{Xendahooklinenumber}%
5601 \newhookcommand@series{Xendbhookinplaceofnumber}%
5602 \newhookcommand@series{Xendahookinplaceofnumber}%
5603
5604 \newhookcommand@series{Xendhangindent}%
5605
5606 \newhooktoggle@series{Xendpagenumberonlyfirst}%
5607 \newhooktoggle@series{Xendpagenumberonlyfirstifsingl}%
5608 \newhooktoggle@series{Xendpagenumberonlyfirstintwo}%
5609 \newhookcommand@series{Xendsympagenum}%
5610 \newhookcommand@series{Xendinplaceofpagenumber}%
5611
5612 \fi
5613 \newhooktoggle@series{Xendlineflag}[apprefwithpage,SErefwithpage]
5614 \newhookcommand@series{Xendt看olines}[apprefwithpage,SErefwithpage]
5615 \newhookcommand@series{Xendmoreethant看olines}[apprefwithpage,SErefwithpage]
5616 \newhooktoggle@series{Xendt看olinesbutnotmore}[apprefwithpage,SErefwithpage]
5617 \newhooktoggle@series{Xendt看olinesonlyinsamepage}[apprefwithpage,
SErefwithpage]
5618 \newhookcommand@series{Xendlinerangeseparator}[apprefwithpage,SErefwithpage
]
5619 \newhookcommand@series{Xendbeforepagenumber}[apprefwithpage,SErefwithpage,
SErefonlypage]
5620 \newhookcommand@series{Xendafterpagenumber}[apprefwithpage,SErefwithpage]
5621 \newhookcommand@series{Xendlineprefixsingle}[apprefwithpage,SErefwithpage]
5622 \newhookcommand@series{Xendlineprefixmore}[apprefwithpage,SErefwithpage]
5623 \newhookcommand@series{Xendsublinesep}[apprefwithpage,SErefwithpage]
5624
5625 %

```

## XXI.4 Hooks for a particular footnote

`\newhooktoggle@specific` `\newhooktoggle@specific` is a generic command to create boolean hook specific to a note.

```

5626 \newcommand{\newhooktoggle@specific}[1]{%
5627   \newtoggle{#1}%
5628   \define@key[mac]{truefootnoteoption}{#1}[]{\global\settogtoggle{#1@}{true}}%
When enabling footnote option
5629   \define@key[mac]{falsefootnoteoption}{#1}[]{\global\settogtoggle{#1@}{false}}
5630 }
5631 %

```

`\newhookarg@specific` `\newhookarg@specific` is a generic command to create argumen hook specific to a note.

```

5632 \newcommand{\newhookarg@specific}[1]{%
5633   \define@key[mac]{truefootnoteoption}{#1}{\global\def\linrangesep@{##1}}%
When enabling footnote option
5634   \define@key[mac]{falsefootnoteoption}{#1}{\global\undef\linrangesep@}%
When
5635 }
5636 %

```

And now, we define some hooks specific to a note.

```

5637 \newhooktoggle@specific{fulllines}%
5638 \newhooktoggle@specific{nonum}
5639 \newhooktoggle@specific{nosep}
5640 \newhookarg@specific{linrangesep}
5641 %

```

`linrangesep@` `\linrangesep@` is defined by the option `linrangesep` of critical notes to change temporarily the line range separator for a specific line. As we have to define it before typesetting the line and undefine it after, we use the family of `xkeyval` package's key.

```

5642 %

```

`\nomk@` `\nomk@` toggle is used by `reledpar` to remove the footnote mark in the text when using `\footnoteXmk`. Read `reledpar` handbook.

```

5643 \newtoggle{nomk@}%
5644 %

```

## XXI.5 Alias

`\Xnolemmaseparator` `\Xnolemmaseparator[⟨series⟩]` is just an alias for `\Xlemmaseparator[⟨series⟩]{}`.

```

5645 \newcommandx*{\Xnolemmaseparator}[1][1]{\Xlemmaseparator{#1}{}}
5646 %

```

## XXII Output routine

Now we begin the output routine and associated things.

### XXII.1 Extra footnotes output

With luck we might only have to change `\@makecol` and `\@reinserts` of the  $\TeX$ '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

`\doxtrafeet` `\doxtrafeet` 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  $\TeX$  are first, then familiar familiar footnotes and finally the critical footnotes.

```

5647 \newcommand*{\l@ddoxtrafeet}{%
5648   \IfStrEq{familiar-critical}{\@fnpos}
5649     {\do@feetX\do@Xfeet}%
5650   {%
5651     \IfStrEq{critical-familiar}{\@fnpos}%
5652       {\do@Xfeet\do@feetX}%
5653     {%
5654       \setbox\@outputbox \vbox{%
5655         \unvbox\@outputbox%
5656         \do@feet@custom@order}{\@fnpos}%
5657       }%
5658     }%
5659   }%
5660 }%
5661 %
5662 %

```

`\do@feet@custom@order` `\do@feet@custom@order` is called when `\@fnpos` is neither 'familiar-critical', nor 'critical-familiar', that is, when the order is more complex. In this case, people must define the order for all footnote series. If they don't,  $\TeX$  could perform an infinite run.

```

5663 \newcommand{\do@feet@custom@order}[2]{%
5664   \def\do##1{%
5665     \edef\@notesseries{\@firstoftwo##1}%
5666     \edef\@notetype{\@secondoftwo##1}%
5667     \ifdefstring{\@notetype}{critical}%
5668       {\csuse{#1append@Xnotes}{\@notesseries}}%
5669       {\ifdefstring{\@notetype}{familiar}%
5670         {\csuse{#1append@notesX}{\@notesseries}}%
5671       }%
5672     }%
5673   }%
5674   \expandafter\docsvlist\expandafter{#2}%
5675 }%

```

```
5676 %
```

**\do@Xfeet** \do@Xfeet is the code extending \@makecol to cater to the extra critical feet.

```
5677 \newcommand*{\do@Xfeet}{%
5678   \setbox\@outputbox \vbox{%
5679     \unvbox\@outputbox
5680     \op@Xfeet}}
5681 %
```

**\@opXfeet** The extra critical feet to be added to the output. . A macro which appends critical notes to the output's routine, also adding vertical space before notes

**\append@Xnotes**

**\print@Xnotes**

```
5682 \newcommand{\append@Xnotes}[1]{%
5683   \ifvoid\csuse{#1footins}\else%
5684     \global\skip\csuse{#1footins}=\csuse{Xbeforenotes@#1}%
5685     \global\advance\skip\csuse{#1footins} by\csuse{Xafterrule@#1}%
5686     \print@Xnotes{#1}%
5687   \fi%
5688 }%
5689 %
```

The normal way to add one series, \print@Xnotes, is replaced by reledpar when using \Pages.

```
5690 \newcommand\print@Xnotes[1]{%
5691   \xdef\@currentseries{#1}%
5692   \csuse{#1footstart}{#1}%
5693   \csuse{#1footgroup}{#1}%
5694 }%
5695 %
```

We print all series of notes by looping on them. We check before printing them that they are not voided.

```
5696 \newcommand*{\@opXfeet}{%
5697   \unless\ifnocritical@%
5698     \gdef\firstXseries@{}%
5699     \def\do##1{%
5700       \append@Xnotes{##1}%
5701     }%
5702     \dolistloop{\@series}%
5703   \fi%
5704 }%
5705 %
```

**\l@ddodoreinxtrafeet** \l@ddodoreinxtrafeet is the code for catering for the extra footnotes within \@reinserts. We use the same category and ordering as in \l@ddoxtrafeet.

```
5706 \newcommand*{\l@ddodoreinxtrafeet}{%
5707   \IfStrEq{familiar-critical}{\@fnpos}
```



```

5708     {\@doreinfeetX\X@doreinfeet}%
5709     {%
5710     \IfStrEq{critical-familiar}{\@fnpos}%
5711         {\X@doreinfeet\@doreinfeetX}%
5712         {\@doreinfeetX\X@doreinfeet}%
5713     }%
5714 }
5715
5716 %

```

`\X@doreinfeet` `\X@doreinfeet` is the code for catering for the extra critical footnotes within `\@reinserts`.

```

5717 \newcommand*{\X@doreinfeet}{%
5718     \unless\ifnocritical@%
5719     \def\do##1{%
5720         \ifvoid\csuse{##1footins}\else%
5721             \insert\csuse{##1footins}{\unvbox\csuse{##1footins}}%
5722         \fi}%
5723     \dolistloop{\@series}
5724     \fi%
5725 }
5726
5727 %

```

`\print@notesX` We have to add all the new kinds of familiar footnotes to the output routine. A macro which appends the familiar footnotes of one series onto the output routine, also adding vertical skip before notes.

`\do@feetX`

```

5728 \@doreinfeetX \newcommand{\append@notesX}[1]{%
5729     \ifvoid\csuse{footins#1}\else%
5730         \global\skip\csuse{footins#1}=\csuse{beforenotesX@#1}%
5731         \global\advance\skip\csuse{footins#1} by\csuse{afterterruleX@#1}%
5732         \print@notesX{#1}%
5733     \fi%
5734 }%
5735 %

```

The normal way to print one series of notes. `\print@Xnotes` is replaced by `reledpar` when using `\Pages`.

```

5736 \newcommand\print@notesX[1]{%
5737     \xdef\@currentseries{#1}%
5738     \csuse{footstart#1}{#1}%
5739     \csuse{footgroup#1}{#1}%
5740 }%
5741 %

```

We print all the series of notes by looping on them. We check before printing them that they are not voided.

```

5742 \newcommand*{\do@feetX}{%

```

```

5743 \unless\ifnofamiliar@%
5744 \gdef\firstseriesX@{%
5745 \setbox\@outputbox \vbox{%
5746 \unvbox\@outputbox%
5747 \def\do##1{%
5748 \append@notesX{##1}%
5749 }%
5750 \dolistloop{\@series}}%
5751 \fi%
5752 }%
5753
5754 \newcommand{\@doreinfeetX}{%
5755 \unless\ifnofamiliar@%
5756 \def\do##1{%
5757 \ifvoid\csuse{footins##1}\else
5758 \insert%
5759 \csuse{footins##1}
5760 {\unvbox\csuse{footins##1}}%
5761 \fi%
5762 }%
5763 \dolistloop{\@series}%
5764 \fi%
5765 }%
5766
5767 %

```

## XXII.2 Patching standard output's commands

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.)

```

5768 \@ifclassloaded{memoir}{%
5769 %
5770 \g@addto@macro{\m@mddoextrafeet}{\l@dddoextrafeet}%
5771 \g@addto@macro{\m@mddodoreinextrafeet}{\l@dddodoreinextrafeet}%
5772 }{%
5773 %

```

memoir is loaded so we use memoir's built in hooks.

memoir has not been loaded, so patch `\@makecol` and `\@reinserts`. If the `fancyhdr` package < version 3.8 has been loaded, we patch the `\latex@makecol` command, because this package redefines the standard `\@makecol` in the preamble, to call `\latex@makecol` which have been `\let` to `\@makecol`. If this package is not loaded, we directly patch `\@makecol`. If the `fancyhdr` package  $\geq$  version 3.8, we also directly patch `\@makecol`, because `fancyhdr` does its own patch `\AtBeginDocument`.

```

5774 \ifboolexpr{%
5775   test{\@ifpackageloaded{fancyhdr}}%
5776   and test {\ifdef{\latex@makecol}}%
5777 }{%
5778   \patchcmd%
5779     {\latex@makecol}%
5780     {\xdef\@freelist{\@freelist\@midlist}}%
5781     {\xdef\@freelist{\@freelist\@midlist}\l@ddoxtrafeet}%
5782     {}%
5783     {\led@error@fail@patch@makecol}%
5784 }{%
5785   \patchcmd%
5786     {\@makecol}%
5787     {\xdef\@freelist{\@freelist\@midlist}}%
5788     {\xdef\@freelist{\@freelist\@midlist}\l@ddoxtrafeet}%
5789     {}%
5790     {\led@error@fail@patch@makecol}%
5791 }%
5792
5793 \patchcmd%
5794   {\@reinserts}%
5795   {\ifvbox}%
5796   {\l@ddodoreinextrafeet\ifvbox}%
5797   {}%
5798   {\led@error@fail@patch@reinserts}%
5799 }
5800
5801 %

```

It turns out that `\@doclearpage` also needs modifying.

`\if@led@nofoot` We have to check if there are any leftover feet.

```

5802 \newif\if@led@nofoot
5803
5804 %

```

```

5805 \@ifclassloaded{memoir}{%
5806 %

```

If the memoir class is loaded we hook into its modified `\@doclearpage`.

```

\@mem@extranofeet \g@addto@macro{\@mem@extranofeet}{%%
5807
5808   \def\do#1{%
5809     \unless\ifnocritical@%
5810       \ifvoid\csuse{#1footins}\else\@mem@nofootfalse\fi%
5811     \fi%
5812     \unless\ifnofamiliar@%
5813       \ifvoid\csuse{footins#1}\else\@mem@nofootfalse\fi%
5814     \fi%

```

```

5815 }
5816 \dolistloop{\@series}%
5817 }%
5818 }%
5819 %

```

As memoir is not loaded we have patch \@doclearpage.

```

\@led@testifnofoot%20 \newcommand*{\@led@testifnofoot}{%
\@doclearpage%21 \@led@nofoottrue%
5822 \ifvoid\footins\else%
5823 \@led@nofootfalse%
5824 \fi%
5825 \def\do##1{%
5826 \unless\ifnocritical@%
5827 \ifvoid\csuse{##1footins}\else%
5828 \@led@nofootfalse%
5829 \fi%
5830 \fi%
5831 \unless\ifnofamiliar@%
5832 \ifvoid\csuse{footins##1}\else%
5833 \@led@nofootfalse%
5834 \fi%
5835 \fi%
5836 }%
5837 \dolistloop{\@series}%
5838 }%
5839
5840 \pretocmd%
5841 {\@doclearpage}%
5842 {\@led@testifnofoot}%
5843 {}%
5844 {\led@error@fail@patch@@doclearpage}%
5845
5846 \patchcmd%
5847 {\@doclearpage}%
5848 {\ifvoid\footins}%
5849 {\if@led@nofoot}%
5850 {}%
5851 {\led@error@fail@patch@@doclearpage}%
5852
5853 }
5854
5855 %

```

## 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 typing `\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.

```
5856 \list@create{\labelref@list}
5857 %
```

`\zz@@@` A convenience macro to zero three labeling counters in one go.

```
5858 \newcommand*{\zz@@@}{000|000|000}% Set three counters to zero in one go
5859
5860 %
```

`\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>33</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>34</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.

```
5861 \newcommand*{\edlabel}[1]{%
5862   \ifl@dpairing\ifautopar%
5863     \strut%
5864   \fi\fi%
5865   \@bsphack%
5866   \ifboolexpr{bool{ledRcol} or bool{ledRcol@}}{%
5867     \ifXnote@%
5868     \protected@write\@auxout{%
5869       {\string\l@dmake@labelsR\space\thepage|\l@dparsedstartline|
5870        \l@dparsedstartsub||\the\c@pstartR|{#1}}}%
5871     \ifdef{\hypertarget}%
5872       {\Hy@raisedlink{\hypertarget{#1}{}}}%
5873   }
```

<sup>33</sup>The remaining macros in this section were kindly revised by Wayne Sullivan, who substantially improved their efficiency and flexibility.

<sup>34</sup>(jdb43@cam.ac.uk) via the ctt thread ‘ledmac cross referencing’, 25 August 2003.

```

5872     {}%
5873   \else%
5874     \write\linenum@outR{\string\@lab}%
5875     \ifx\labelref@listR\empty%
5876       \xdef\label@refs{\zz@@@}%
5877     \else%
5878       \glp\labelref@listR\to\label@refs%
5879     \fi%
5880     \ifvmode%
5881       \advancelabel@refs%
5882     \fi%
5883 %

```

Use code from the kernel `\label` command to write the correct page number. Also define an `hypertarget` if `hyperref` package is loaded.

```

5884   \protected@write\@auxout{}%
5885   {\string\l@dmake@labelsR\space\thepage|\label@refs|\the\c@pstartR
|{\#1}}%
5886   \ifdef{\hypertarget}%
5887     {\Hy@raisedlink{\hypertarget{\#1}{}}}%
5888     {}%
5889   \fi%
5890 }{%
5891   \ifXnote%
5892     \ifl@dpairing%pstart or pstartL?
5893     \protected@write\@auxout{}%
5894     {\string\l@dmake@labels\space\thepage|\l@dparsedstartline|\
l@dparsedstartsub||\the\c@pstartL|{\#1}}%
5895     \ifdef{\hypertarget}%
5896       {\Hy@raisedlink{\hypertarget{\#1}{}}}%
5897       {}%
5898     \else%
5899       \protected@write\@auxout{}%
5900       {\string\l@dmake@labels\space\thepage|\l@dparsedstartline|\
l@dparsedstartsub||\the\c@pstart|{\#1}}%
5901       \ifdef{\hypertarget}%
5902         {\Hy@raisedlink{\hypertarget{\#1}{}}}%
5903         {}%
5904       \fi%
5905     \else%
5906       \write\linenum@out{\string\@lab}%
5907       \ifx\labelref@list\empty%
5908         \xdef\label@refs{\zz@@@}%
5909       \else%
5910         \glp\labelref@list\to\label@refs%
5911       \fi%
5912       \ifvmode%
5913         \advancelabel@refs%
5914       \fi%
5915       \ifl@dpairing%Pstart or PstartL?

```

```

5916 \protected@write\@auxout{}%
5917 {\string\l@dmake@labels\space\thepage|\label@refs|\the\c@pstartL
|{#1}}}%
5918 \ifdef\hypertarget{%
5919 {\Hy@raisedlink{\hypertarget{#1}{}}}%
5920 {}%
5921 \else%
5922 \protected@write\@auxout{}%
5923 {\string\l@dmake@labels\space\thepage|\label@refs|\the\c@pstart
|{#1}}}%
5924 \ifdef\hypertarget{%
5925 {\Hy@raisedlink{\hypertarget{#1}{}}}%
5926 {}%
5927 \fi%
5928 \fi%
5929 }%
5930 \@esphack}%
5931 %
5932 %

```

`\advancelabel@refs` `\labelrefsparseline` `\labelrefsparsesubline` In cases where `\edlabel` is the first element in a paragraph, we have a problem with line counts, because line counts change only at the first horizontal box of the paragraph. Hence, we need to test `\edlabel` if it occurs at the start of a paragraph. To do so, we use `\ifvmode`. If the test is true, we must advance by one unit the amount of text we write into the `.aux` file. We do so using `\advancelabel@refs` command.

```

5933 \newcounter{line}%
5934 \newcounter{subline}%
5935 \newcounter{absline}%
5936 \newcommand{\advancelabel@refs}{%
5937 \setcounter{line}{\expandafter\labelrefsparseline\label@refs}%
5938 \stepcounter{line}%
5939 \setcounter{absline}{\expandafter\labelrefsparsabsline\label@refs}%
5940 \stepcounter{absline}%
5941 \ifsublines@%
5942 \setcounter{subline}{\expandafter\labelrefsparsesubline\label@refs}%
5943 %
5944 \stepcounter{subline}{1}%
5945 \def\label@refs{\theline|\thesubline|\theabsline}%
5946 \else%
5947 \def\label@refs{\theline|0|\theabsline}%
5948 \fi%
5949 }
5950 \def\labelrefsparseline#1|#2|#3{#1}%
5951 \def\labelrefsparsesubline#1|#2|#3{#2}%
5952 \def\labelrefsparsabsline#1|#2|#3{#3}%
5953 %

```

`\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 absolute line number, #5 pstart number, #6 label.

```

5953 \newcommand*\l@dmake@labels{}
5954 \def\l@dmake@labels#1|#2|#3|#4|#5|#6{%
5955   \expandafter\ifx\csname the@label\csuse{XR@prefix}#6\endcsname%
5956     \relax%
5957   \else%
5958     \led@warn@DuplicateLabel{\csuse{XR@prefix}#6}%
5959   \fi
5960   \global\providetoggle{label@#6@ledRcol}%False is the default value of
this toggle, which tell us if a label is linked to a right or a left side
5961   \expandafter\gdef\csname the@label\csuse{XR@prefix}#6\endcsname
{#1|#2|#3|#4|#5|\relax}%
5962   \ignorespaces}
5963
5964 %

```

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.

```

5965 \AtBeginDocument{%
5966   \def\l@dmake@labels#1|#2|#3|#4|#5|#6{%
5967 }
5968
5969 %

```

**\@lab** The `\@lab` command, which appears in the `\linenum@out` file, appends the current value of page, line, sub-line, and absolute 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`.

```

5970
5971 \newcommand*\@lab{%
5972   \ifledRcol
5973     \xright@appenditem{\linenumr@p{\line@numR}}|%
5974     \ifsublines@ \sublinenumr@p{\subline@numR}\else 0\fi|\the\
absline@numR}%
5975     \to\labelref@listR
5976   \else
5977     \xright@appenditem{\linenumr@p{\line@num}}|%

```



```

5978     \ifsublines@ \sublinenumr@p{\subline@num}\else 0\fi|\the\absline@num}
5979 %
5980     \to\labelref@list
5981 \fi}
5982 %

```

**\applabel** \applabel, if called in \edtext will insert automatically both a start and an end label for the current edtext lines.

```

5982 \newcommand*{\applabel}[1]{%
5983     \if@edtext@secondarg%
5984 %

```

Label should not be already defined.

```

5985     \ifcsundef{the@label#1}{%
5986         \csdef{the@label#1}{\applabel}%
5987     }%
5988     {%
5989         \led@warn@DuplicateLabel{#1 (\applabel)}%
5990     }%
5991 %

```

Parse the \edtext line numbers.

```

5992     \expandafter\l@dp@rsefootspec\l@d@nums| %
5993 %

```

Use the  $\TeX$  standard hack for label.

```

5994     \@bsphack%
5995 %

```

And now, write the data in the auxiliary file.

```

5996     \ifledRcol%
5997         \protected@write\@auxout{}%
5998             {\string\l@dmake@labelsR\space\l@d@p@rsestartpage|\
6000 \l@d@p@rsestartline|\l@d@p@rsestartsub||\the\c@pstartR|{#1:start}}}%
6001         \ifdef{\hypertarget}%
6002             {\Hy@raisedlink{\hypertarget{#1:start}}{}}%
6003             {}%
6004         \protected@write\@auxout{}%
6005             {\string\l@dmake@labelsR\space\l@d@p@rseendpage|\l@d@p@rseendline
6006 ||\l@d@p@rseendsub|\the\c@pstartR|{#1:end}}}%
6007         \else%
6008             \ifl@d@p@rsepairing%pstart or pstartL?
6009             \protected@write\@auxout{}%
6010                 {\string\l@dmake@labelsR\space\l@d@p@rsestartpage|\
\l@d@p@rsestartline|\l@d@p@rsestartsub||\the\c@pstartL|{#1:start}}}%
6011                 \ifdef{\hypertarget}%
6012                     {\Hy@raisedlink{\hypertarget{#1:start}}{}}%
6013                     {}%

```

```

6011         \protected@write\@auxout{}%
6012         {\string\l@dmake@labels\space\l@dparsedendpage|\
l@dparsedendline|\l@dparsedendsub||\the\c@pstartL|{\#1:end}}%
6013         \else%
6014         \protected@write\@auxout{}%
6015         {\string\l@dmake@labels\space\l@dparsedstartpage|\
l@dparsedstartline|\l@dparsedstartsub||\the\c@pstart|{\#1:start}}%
6016         \ifdef{\hypertarget}%
6017         {\Hy@raisedlink{\hypertarget{\#1:start}}}%
6018         {}%
6019         \protected@write\@auxout{}%
6020         {\string\l@dmake@labels\space\l@dparsedendpage|\
l@dparsedendline|\l@dparsedendsub||\the\c@pstart|{\#1:end}}%
6021         \fi%
6022     \fi%
6023 %

```

Use the  $\TeX$  standard hack for label.

```

6024     \@esphack%
6025 %

```

Warning if `\applabel` is called outside of `\edtext`.

```

6026     \else%
6027     \led@warn@AppLabelOutSecondArgEdtext{\#1}%
6028     \fi%
6029 %

```

End of `\applabel`

```

6030 }%
6031 %

```

`\edlabels` `\edlabelS` and `\edlabelE` are just used to mark the beginning and the end of a passage.

```

\edlabelE
\edlabelSE
6032 \newcommand{\edlabelS}[1]{%
6033     \edlabel{\#1:start}%
6034 }
6035 \newcommand{\edlabelE}[1]{%
6036     \edlabel{\#1:end}%
6037 }
6038 \newcommand{\edlabelSE}[1]{%
6039     \edlabelS{\#1}%
6040     \edlabelE{\#1}%
6041 }
6042 %

```

`\wrap@edcrossref` `\wrap@edcrossref` is called around all `reledmac` crossref commands, except those which start with `x`. It adds the hyperlink.

```

6043 \newrobustcmd{\wrap@edcrossref}[2]{%
6044   \ifdef{\hyperlink}%
6045     {\hyperlink{#1}{#2}}%
6046     {#2}%
6047 }
6048 %

```

**\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.

```

6049 \newcommand*\edpageref[1]{\l@dref@undefined{#1}\wrap@edcrossref{#1}{\l@dgetref@num{1}{#1}}}
6050 \newcommand*\xpageref[1]{\l@dgetref@num{1}{#1}}
6051
6052 %

```

**\edlineref** If the specified label exists, \lineref gives its line number.

**\xlineref**

```

6053 \newcommand*\edlineref[1]{%
6054   \l@dref@undefined{#1}%
6055   \wrap@edcrossref{#1}{%
6056     \providetoggle{label@#1@ledRcol}%Required for the first run, when the
        label has not yet been parsed on the .aux file
6057     \iftoggle{label@#1@ledRcol}%
6058       {\linenumrepR{\l@dgetref@num{2}{#1}}}%
6059       {\linenumrep{\l@dgetref@num{2}{#1}}}%
6060     \xflagref{#1}%
6061   }%
6062 }%
6063 \newcommand*\xlineref[1]{\l@dgetref@num{2}{#1}}%
6064
6065 %

```

**\sublineref** If the specified label exists, \sublineref gives its sub-line number.

**\xsublineref**

```

6066 \newcommand*\sublineref[1]{%
6067   \l@dref@undefined{#1}%
6068   \wrap@edcrossref{#1}{%
6069     \providetoggle{label@#1@ledRcol}%Required for the first run, when the
        label has not yet been parsed on the .aux file
6070     \iftoggle{label@#1@ledRcol}%
6071       {\sublinenumrepR{\l@dgetref@num{3}{#1}}}%
6072       {\sublinenumrep{\l@dgetref@num{3}{#1}}}%

```

```

6073 }%
6074 }%
6075 \newcommand*\xsublineref}[1]{\l@getref@num{3}{#1}}
6076
6077 %

```

**\xabslineref** If the specified label exists, \xabslineref gives its absolute line number. That is used usually only by some reledmac internal macros.

```

6078 \newcommand*\xabslineref}[1]{\l@getref@num{4}{#1}}%
6079 %

```

**\pstartref** If the specified label exists, \pstartref gives its pstart number.

```

\pstartref
\xpstartref
6080 \newcommand*\pstartref}[1]{\l@dref@undefined{#1}\wrap@edcrossref{#1}{\l@getref@num{5}{#1}}}%
6081 \newcommand*\xpstartref}[1]{\l@getref@num{5}{#1}}%
6082
6083 %

```

**\xflagref** \xflagref finds the side flag of any ref defined with \edlabel.

```

6084 \newcommand*\xflagref}[1]{\l@getref@num{6}{#1}}%
6085 %

```

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.

```

6086 \newcommand*\l@dref@undefined}[1]{%
6087   \expandafter\ifx\csname the@label#1\endcsname\relax
6088   \led@warn@RefUndefined{#1}%
6089   \fi}
6090
6091 %

```

**\l@getref@num** Next, \l@getref@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), (4) pstart number or (5) side flag. (This switching is done by calling \l@dlabel@parse.) The second argument is the label-macro, which because of the \@lab macro above is defined to be a string of the type 123|456|789.

```

6092 \newcommand*\l@getref@num}[2]{%
6093   \expandafter
6094   \ifx\csname the@label#2\endcsname \relax

```

```

6095     000%
6096     \else
6097         \expandafter\expandafter\expandafter
6098         \l@dlabel@parse\csname the@label#2\endcsname|#1%
6099     \fi}
6100
6101 %

```

**\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, 4 or 5) which defines which of the earlier six numbers to pick out. (It was earlier given as the first argument of \l@dgetref@num.)

```

6102 \newcommand*\l@dlabel@parse{-}{}
6103 \def\l@dlabel@parse#1|#2|#3|#4|#5|#6|#7{%
6104     \ifcase #7%
6105         \or #1%
6106         \or #2%
6107         \or #3%
6108         \or #4%
6109         \or #5%
6110         \or #6%
6111     \fi}
6112 %

```

**\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 cannot 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.

```

6113 \newcommand*\xxref}[2]{%
6114     {%
6115         \expandafter\ifx\csname the@label#1\endcsname \relax%
6116             \expandafter\let\csname the@@label#1\endcsname\zz@@@%
6117         \else%
6118             \expandafter\def\csname the@@label#1\endcsname{\l@dgetref@num
6119             {1}{#1}|\l@dgetref@num{2}{#1}|\l@dgetref@num{3}{#1}}%
6120             \fi%
6121         \expandafter\ifx\csname the@label#2\endcsname \relax%
6122             \expandafter\let\csname the@@label#2\endcsname\zz@@@%
6123         \else%

```

```

6123 \expandafter\def\csname the@@label#2\endcsname{\l@getref@num
        {1}{#2}|\l@getref@num{2}{#2}|\l@getref@num{3}{#2}}%
6124 \fi%
6125 \letcs{\@tempa}{the@@label#1}%
6126 \letcs{\@tempb}{the@@label#2}%
6127 \global\appto\@beforeinsertofthisedtext{\def\@this@crossref@start{#1}}%
6128 \global\appto\@beforeinsertofthisedtext{\def\@this@crossref@end{#2}}%
6129 \linenum{\@tempa|%
6130 \@tempb}}}%
6131
6132 %

```

`\appref` `\SEref`, `\apprefwithpage`, `\SErefwithpage` and `\SEonlypage` print cross-ref to some start / end lines defined by specific commands. It prints the lines as they should be printed in the apparatus (critical notes for not suffixed versions, endnotes for suffixed versions).

`\SErefwithpage` Here we define hooks similar to some those related to critical footnotes or endnotes.

`\SErefwithpage` So, first declare the default value of the hooks for the pseudo-series. Also declare the internal toggle which are switch by `reledmac`.

```

6133 \def\Xtwolines@appref{}%
6134 \def\Xtwolines@SEref{}%
6135
6136 \def\Xmorethantwolines@appref{}%
6137 \def\Xmorethantwolines@SEref{}%
6138
6139 \def\Xlinerangeseparator@appref{\endashchar}%
6140 \def\Xlinerangeseparator@SEref{\endashchar}%
6141
6142 \def\Xsublinesep@appref{\fullstop}%
6143 \def\Xsublinesep@SEref{\fullstop}%
6144
6145 \def\Xpagelinesep@appref{\fullstop}%
6146 \def\Xpagelinesep@SEref{\fullstop}%
6147
6148
6149 \newtoggle{Xtwolinesbutnotmore@appref}%
6150 \newtoggle{Xtwolinesbutnotmore@SEref}%
6151
6152 \newtoggle{Xtwolinesonlyinsamepage@appref}%
6153
6154 \newtoggle{Xtwolinesonlyinsamepage@SEref}%
6155
6156 \newtoggle{Xlineflag@appref}%
6157 \toggletrue{Xlineflag@appref}%Here exception
6158 \newtoggle{Xlineflag@SEref}%
6159 \toggletrue{Xlineflag@SEref}%%Here exception
6160
6161 \def\Xendtwolines@apprefwithpage{}%

```

```

6162 \def\Xendtwolines@SErefwithpage{}%
6163
6164 \def\Xendmoreethantwolines@apprefwithpage{}%
6165 \def\Xendmoreethantwolines@SErefwithpage{}%
6166
6167 \def\Xendlinrangeseparator@apprefwithpage{\endashchar}
6168 \def\Xendlinrangeseparator@SErefwithpage{\endashchar}
6169 \def\Xendlinrangeseparator@SErefonlypage{\endashchar}
6170
6171 \def\Xendbeforepagenumber@apprefwithpage{p.}%
6172 \def\Xendbeforepagenumber@SErefwithpage{p.}%
6173 \def\Xendbeforepagenumber@SEonlypage{p.}%
6174
6175 \def\Xendafterpagenumber@apprefwithpage{ }%
6176 \def\Xendafterpagenumber@SErefwithpage{ }%
6177
6178
6179 \def\Xendlineprefixsingle@apprefwithpage{}%
6180 \def\Xendlineprefixsingle@SErefwithpage{}%
6181
6182 \def\Xendlineprefixmore@apprefwithpage{}%
6183 \def\Xendlineprefixmore@SErefwithpage{}%
6184
6185 \newtoggle{Xendtwolinesbutnotmore@apprefwithpage}%
6186 \newtoggle{Xendtwolinesbutnotmore@SErefwithpage}%
6187
6188 \def\Xendsublinesep@apprefwithpage{\fullstop}%
6189 \def\Xendsublinesep@SErefwithpage{\fullstop}%
6190
6191 \newtoggle{Xendtwolinesonlyinsamepage@apprefwithpage}%
6192 \newtoggle{Xendtwolinesonlyinsamepage@SErefwithpage}%
6193
6194 \newtoggle{Xendlineflag@apprefwithpage}
6195 \toggletrue{Xendlineflag@apprefwithpage}%Here, exception
6196 \newtoggle{Xendlineflag@SErefwithpage}
6197 \toggletrue{Xendlineflag@SErefwithpage}%Here, exception
6198
6199 %

```

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.

```

6200
6201 \gdef\Xboxstartlinenum@appref{Opt}
6202 \gdef\Xboxstartlinenum@SEref{Opt}
6203
6204 \gdef\Xboxendlinenum@appref{Opt}
6205 \gdef\Xboxendlinenum@SEref{Opt}
6206
6207 \gdef\Xendboxstartlinenum@apprefwithpage{Opt}

```

```

6208 \gdef\Xendboxstartlinenum@SErefwithpage{Opt}
6209
6210 \gdef\Xendboxendlinenum@apprefwithpage{Opt}
6211 \gdef\Xendboxendlinenum@SErefwithpage{Opt}
6212
6213 \newtoggle{Xendpagenumberonlyfirst@apprefwithpage}
6214 \newtoggle{Xendpagenumberonlyfirst@SErefwithpage}
6215
6216 \newtoggle{Xendpagenumberonlyfirstifsingle@apprefwithpage}
6217 \newtoggle{Xendpagenumberonlyfirstifsingle@SErefwithpage}
6218
6219 \newtoggle{Xendpagenumberonlyfirstintwo@apprefwithpage}
6220 \newtoggle{Xendpagenumberonlyfirstintwo@SErefwithpage}
6221
6222 \gdef\Xendsympagenum@apprefwithpage{}
6223 \gdef\Xendsympagenum@SErefwithpage{}
6224
6225 \gdef\Xendinplaceofpagenumber@apprefwithpage{}
6226 \gdef\Xendinplaceofpagenumber@SErefwithpage{}
6227
6228 %

```

Now, declare the default values of \@apprefprefixsingle and \@apprefprefixmore, \@SErefprefix, \@SErefprefixmore and the commands which defines them.

```

6229 \newcommand\@apprefprefixsingle{}%
6230 \newcommand\@SErefprefixsingle{}%
6231
6232 \newcommand\@apprefprefixmore{}%
6233 \newcommand\@SErefprefixmore{}%
6234
6235 \newcommand{\setapprefprefixsingle}[1]{%
6236   \gdef\@apprefprefixsingle{#1}%
6237 }
6238 \newcommand{\setSErefprefixsingle}[1]{%
6239   \gdef\@SErefprefixsingle{#1}%
6240 }
6241
6242 \newcommand{\setapprefprefixmore}[1]{%
6243   \gdef\@apprefprefixmore{#1}%
6244 }
6245 \newcommand{\setSErefprefixmore}[1]{%
6246   \gdef\@SErefprefixmore{#1}%
6247 }
6248
6249 %

```

And not \setSErefonlypageprefixsingle and \setSErefonlypageprefixmore.

```

6250 \newcommand{\setSErefonlypageprefixsingle}[1]{%
6251   \gdef\SErefonlypage@prefixsingle{#1}%
6252 }%

```



```

6253 \newcommand{\setSErefonlypageprefixmore}[1]{%
6254   \gdef\SErefonlypage@prefixmore{#1}%
6255 }%
6256 %

```

And now, the main commands: \appref, \apprefwithpage, \SEref and \SErefwithpage. These commands call \reformatted@ and \reformattedwithpage, which calls \printlines and \printendlines. That is why we have previously declared all hooks values tested inside these last commands.

```

6257
6258 \newcommandx{\appref}[2][1,usedefault]{\reformatted@{#1}{#2}{appref}}
6259 \newcommandx{\SEref}[2][1,usedefault]{\reformatted@{#1}{#2}{SEref}}
6260
6261 \newcommandx{\apprefwithpage}[2][1,usedefault]{\reformattedwithpage@
6262 {#1}{#2}{appref}}
6263 \newcommandx{\SErefwithpage}[2][1,usedefault]{\reformattedwithpage@
6264 {#1}{#2}{SEref}}
6265 \newcommandx{\SErefonlypage}[2][1,usedefault]{\reformattedonlypage@
6266 {#1}{#2}{SEref}}
6267
6268 \newcommand{\reformatted@}[3]{%
6269   \def\do##1{%
6270     \setkeys[mac]{truefootnoteoption}{##1}%
6271   }%
6272   \notblank{#1}{\docsvlist{#1}}{%
6273     \xdef\@currentseries{#3}%
6274     \ifcempty{@#3prefixmore}%
6275       {\@apprefprefixsingle}%
6276       {%
6277         \IfEq{\xlineref{#2:start}}{\xlineref{#2:end}}{%
6278           {\csuse{@#3prefixsingle}}%
6279           {\csuse{@#3prefixmore}}%
6280         }%
6281       }%
6282   }%
6283   \ifboolexpr{%
6284     test{\ifcsundef{the@label#2:start}}%
6285     or test{\ifcsundef{the@label#2:end}}%
6286   }%
6287   {\led@warn@pairRefUndefined{#2}\nfss@text{\reset@font\bfseries ??}}%
6288   {%
6289     \def\@this@crossref@start{#2:start}%
6290     \def\@this@crossref@end{#2:end}%
6291     \printlines\xpageref{#2:start}|\xlineref{#2:start}|\xsublineref{#2:
6292 start}|\xpageref{#2:end}|\xlineref{#2:end}|\xsublineref{#2:end}|\relax|\
xflagref{#2:start}|}%
6293     \undef\@this@crossref@end%
6294     \undef\@this@crossref@start%
6295   }%
6296   \def\do##1{%
6297     \setkeys[mac]{falsefootnoteoption}{##1}%

```

```

6293 }%
6294 \notblank{#1}{\docsvlist{#1}}{}%
6295 }%
6296
6297 \newcommand{\reformattedwithpage@}[3]{%
6298   \def\do##1{%
6299     \setkeys[mac]{truefootnoteoption}{##1}%
6300   }%
6301   \notblank{#1}{\docsvlist{#1}}{}%
6302   \xdef\@currentseries{#3withpage}%
6303   \ifboolexpr{%
6304     test{\ifcsundef{the@label#2:start}}%
6305     or test{\ifcsundef{the@label#2:end}}%
6306   }%
6307   {\led@warn@pairRefUndefined{#2}\nfss@text{\reset@font\bfseries ??}}%
6308   {%
6309     \def\@this@crossref@start{#2:start}%
6310     \def\@this@crossref@end{#2:end}%
6311     \printendlines\xpageref{#2:start}|\xlineref{#2:start}|\xsublineref{#2:
start}|\xpageref{#2:end}|\xlineref{#2:end}|\xsublineref{#2:end}|\relax|\
xflagref{#2:start}|}%
6312     \undef\@this@crossref@end%
6313     \undef\@this@crossref@start%
6314   }%
6315   \def\do##1{%
6316     \setkeys[mac]{falsefootnoteoption}{##1}%
6317   }%
6318   \notblank{#1}{\docsvlist{#1}}{}%
6319 }%
6320
6321 \newcommand{\reformattedonlypage@}[3]{%
6322   \def\do##1{%
6323     \setkeys[mac]{truefootnoteoption}{##1}%
6324   }%
6325   \notblank{#1}{\docsvlist{#1}}{}%
6326   \xdef\@currentseries{#3onlypage}%
6327   \ifboolexpr{%
6328     test{\ifcsundef{the@label#2:start}}%
6329     or test{\ifcsundef{the@label#2:end}}%
6330   }%
6331   {\led@warn@pairRefUndefined{#2}\nfss@text{\reset@font\bfseries ??}}%
6332   {\ifnumequal{\xpageref{#2:end}}{\xpageref{#2:start}}%
6333     {%
6334       \ifcsvoid{#3onlypage@prefixsingle}%
6335       {}%
6336       {\csletcs{Xendbeforepagenumber@#3onlypage}{#3onlypage@prefixsingle
}}%
6337       \printnpnum{%
6338         \wrap@edcrossref{#2:start}{\xpageref{#2:start}}%
6339       }%

```

```

6340 }%
6341 {%
6342 \ifcsvoid{#3onlypage@prefixmore}%
6343 {}%
6344 {\csletcs{Xendbeforepagenumber@#3onlypage}{#3onlypage@prefixmore}}%
6345 \ifdefined\linangesep%
6346 \printnpnum{%
6347 \wrap@edcrossref{#2:start}{\xpageref{#2:start}}%
6348 \linangesep%
6349 \wrap@edcrossref{#2:end}{\xpageref{#2:end}}%
6350 }%
6351 \else%
6352 \printnpnum{%
6353 \wrap@edcrossref{#2:start}{\xpageref{#2:start}}%
6354 \csuse{Xendlinangeseparator@}\@currentseries%
6355 \wrap@edcrossref{#2:end}{\xpageref{#2:end}}%
6356 }%
6357 \fi%
6358 }%
6359 }%
6360 \def\do##1{%
6361 \setkeys[mac]{falsefootnoteoption}{##1}%
6362 }%
6363 \notblank{#1}{\docsvlist{#1}}{}%
6364 }%
6365 %

```

`\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 insert `\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`.

```

6366 \newcommand*{\edmakelabel}[2]{\expandafter\def\csname the@label#1\
6367 endcsname{#2}}
6368 %

```

(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. 132 and V.9 p. 99), since `\xxref` makes a call to `\linenum` in order to do its work.)

## XXIII.1 Compatibility with xref

Here, we provide compatibility with the `xref` to enable `reledmac`’s cross-referencing to external documents. We assume that the user loads `xref` *before* `reledmac`, but uses `\externaldocument` *after* loading `reledmac`.

**\XR@test** First, we patch the `xr` macro `\XR@test`, which is called on every line of the external `.aux` file, in order to also call macros specific to `reledmac`.

```
6369 \pretocmd{\XR@test}%
6370   {\XR@test@mac+++#1#2#3#4+++}%
6371   {}%
6372   {}%
6373 %
```

**\XR@test@mac** The `\XR@test@mac` takes the full content of a line of the external `.aux` files, with the three final dots added by `xr`.

```
6374 \long\def\xR@test@mac+++#1+++{\XR@test@mac@test#1}
6375 %
```

**\XR@test@mac@test** And finally, `\XR@test@mac@test` does the job. This code is based on the `\XR@test` macro of the `xr` package. However, note that the `\XR@prefix` is not called here, but it is integrated directly in `\l@dmake@labels` and `\l@dmake@labelsR`.

```
6376 \long\def\xR@test@mac@test#1#2...{%The triple dots (NOT \ldots) are because
6377   of the line 22 of xr.sty v5.02 1994/05/28
6378   \ifx#1\l@dmake@labels%
6379     \l@dmake@labels#2%
6380   \else
6381     \ifx#1\l@dmake@labelsR%
6382       \l@dmake@labelsR #2%
6383     \fi%
6384   \fi%
6385   }%
6386 %
```

## 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.

```
6386 \pretocmd{\@xympar}%
6387   {\ifnumberedpar@
6388     \led@warn@NoMarginpars
6389     \@esphack
6390   \else}%
6391   {}%
6392   {}%
6393 %
6394 \apptocmd{\@xympar}%
6395   {\fi}%
```

```

6396 {}
6397 {}
6398
6399 %

```

We provide side notes as replacement for `\marginpar` in numbered text.

`\sidenote@margin` These are the sidenote equivalents to `\line@margin` and `\linenummargin` for specifying which margin. The default is the right margin (opposite to the default for line numbers). `\l@dgetsidenote@margin` returns the number associated to side note margin:

**left:** 0

**right:** 1

**outer:** 2

**inner:** 3

```

6400 \newcount\sidenote@margin
6401 \newcommand*{\sidenotemargin}[1]{\%
6402   \l@dgetsidenote@margin{#1}%
6403   \ifnum\@l@dttempcntb>\m@ne
6404     \ifledRcol
6405       \global\sidenote@marginR=\@l@dttempcntb
6406     \else
6407       \global\sidenote@margin=\@l@dttempcntb
6408     \fi
6409   \fi}}
6410 \newcommand*{\l@dgetsidenote@margin}[1]{\%
6411   \def\@tempa{#1}\def\@tempb{left}%
6412   \ifx\@tempa\@tempb
6413     \@l@dttempcntb \z@
6414   \else
6415     \def\@tempb{right}%
6416     \ifx\@tempa\@tempb
6417       \@l@dttempcntb \@ne
6418     \else
6419       \def\@tempb{outer}%
6420       \ifx\@tempa\@tempb
6421         \@l@dttempcntb \tw@
6422       \else
6423         \def\@tempb{inner}%
6424         \ifx\@tempa\@tempb
6425           \@l@dttempcntb \thr@@
6426         \else
6427           \led@warn@BadSidenotemargin
6428           \@l@dttempcntb \m@ne
6429         \fi

```

```

6430     \fi
6431     \fi
6432     \fi}
6433 \sidenotemargin{right}
6434
6435 %

```

`\l@dlp@rbox` We need two boxes to store sidenote texts.

```

\l@drp@rbox
6436 \newbox\l@dlp@rbox
6437 \newbox\l@drp@rbox
6438
6439 %

```

`\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
6440 \newdimen\ledlsnotewidth \ledlsnotewidth=\marginparwidth
\ledrsnotesep
6441 \newdimen\ledrsnotewidth \ledrsnotewidth=\marginparwidth
\ledlsnotefontsetup
6442 \newdimen\ledlsnotesep \ledlsnotesep=\linenumsep
\ledrsnotefontsetup
6443 \newdimen\ledrsnotesep \ledrsnotesep=\linenumsep
6444 \newcommand*\ledlsnotefontsetup{\raggedleft\footnotesize}
6445 \newcommand*\ledrsnotefontsetup{\raggedright\footnotesize}
6446
6447 %

```

`\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
6448 \newcommand*\ledleftnote[1]{\edtext{\l@dlsnote{#1}}}
6449 \newcommand*\ledrightnote[1]{\edtext{\l@drsnote{#1}}}
6450 \newcommand*\ledsidenote[1]{\edtext{\l@dcnote{#1}}}%
6451 \newcommand*\ledinnernote[1]{\edtext{\l@disnote{#1}}}%
6452 \newcommand*\ledouternote[1]{\edtext{\l@dosnote{#1}}}%
6453 %

```

`\l@dlsnote` . The ‘footnotes’ for left, right, and moveable sidenotes. The whole scheme is reminiscent  
`\l@drsnote` of the critical footnotes code.

```

\l@dcnote
6454 \newif\ifrighnoteup
\l@desnote
6455 \righnoteuptrue
\l@disnote
6456
6457 \newcommand*\l@dlsnote[1]{%
6458   \begingroup%
6459   \newcommand{\content}{#1}%
6460   \ifnumberedpar@
6461     \ifledRcol%
6462       \xright@appenditem{\noexpand\l@dlsnote{\expandonce\content}}%

```

```

6463         \to\inserts@listR
6464         \global\advance\insert@countR \@ne%
6465     \else%
6466         \xright@appenditem{\noexpand\vl@dlsnote{\expandonce\content}}}%
6467         \to\inserts@list
6468         \global\advance\insert@count \@ne%
6469     \fi
6470 \fi%
6471 \ignorespaces%
6472 \endgroup%
6473 }%
6474
6475 \newcommand*{\l@drsnote}[1]{%
6476     \begingroup%
6477     \newcommand{\content}{#1}%
6478     \ifnumberedpar@
6479         \ifledRcol%
6480             \xright@appenditem{\noexpand\vl@drsnote{\expandonce\content}}}%
6481             \to\inserts@listR
6482             \global\advance\insert@countR \@ne%
6483         \else%
6484             \xright@appenditem{\noexpand\vl@drsnote{\expandonce\content}}}%
6485             \to\inserts@list
6486             \global\advance\insert@count \@ne%
6487         \fi
6488     \fi\ignorespaces%
6489 \endgroup%
6490 }%
6491
6492 \newcommand*{\l@dcsnote}[1]{%
6493     \begingroup%
6494     \newcommand{\content}{#1}%
6495     \ifnumberedpar@
6496         \ifledRcol%
6497             \xright@appenditem{\noexpand\vl@dcsnote{\expandonce\content}}}%
6498             \to\inserts@listR
6499             \global\advance\insert@countR \@ne%
6500         \else%
6501             \xright@appenditem{\noexpand\vl@dcsnote{\expandonce\content}}}%
6502             \to\inserts@list
6503             \global\advance\insert@count \@ne%
6504         \fi
6505     \fi\ignorespaces%
6506 \endgroup%
6507 }%
6508
6509 \newcommand*{\l@disnote}[1]{%
6510     \begingroup%
6511     \newcommand{\content}{#1}%
6512     \ifnumberedpar@

```

```

6513 \ifledRcol%
6514 \xright@appenditem{\noexpand\vl@disnote{\expandonce\content}}}%
6515 \to\inserts@listR%
6516 \global\advance\insert@countR \@ne%
6517 \else%
6518 \xright@appenditem{\noexpand\vl@disnote{\expandonce\content}}}%
6519 \to\inserts@list%
6520 \global\advance\insert@count \@ne%
6521 \fi%
6522 \fi\ignorespaces%
6523 \endgroup%
6524 }%
6525
6526 \newcommand*{\l@dosnote}[1]{%
6527 \begingroup%
6528 \newcommand{\content}{#1}%
6529 \ifnumberedpar%
6530 \ifledRcol%
6531 \xright@appenditem{\noexpand\vl@dosnote{\expandonce\content}}}%
6532 \to\inserts@listR%
6533 \global\advance\insert@countR \@ne%
6534 \else%
6535 \xright@appenditem{\noexpand\vl@dosnote{\expandonce\content}}}%
6536 \to\inserts@list%
6537 \global\advance\insert@count \@ne%
6538 \fi%
6539 \fi\ignorespaces%
6540 \endgroup%
6541 }%
6542
6543 %

```

**\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 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 **\vl@disnote** to do, in order to:

- Store the content of **\ledsidenote** to **\l@dcsnotetext** in any cases.
- Store the content of **\rightsidenote** to:
  - **\l@dcsnotetext** if **\ledsidenote** is to be put on right.
  - **\l@dcsnotetext@r** if **\ledsidenote** is to be put on left.
- Store the content of **\leftsidenote** to:
  - **\l@dcsnotetext** if **\ledsidenote** is to be put on left.
  - **\l@dcsnotetext@l** if **\ledsidenote** is to be put on right.



`\vl@disnote` and `\vl@dosnote` just call `\vl@dlsnote` or `\vl@drsnote`, depending of the page.

```

6544 \newcommand*{\vl@dlsnote}[1]{%
6545   \ifledRcol{%
6546     \@l@tempcntb=\sidenote@marginR%
6547     \ifnum\@l@tempcntb>\@ne%
6548       \advance\@l@tempcntb by\page@numR%
6549     \fi%
6550   \else%
6551     \@l@tempcntb=\sidenote@margin%
6552     \ifnum\@l@tempcntb>\@ne%
6553       \advance\@l@tempcntb by\page@num%
6554     \fi%
6555   \fi%
6556   \ifodd\@l@tempcntb%
6557     \listgadd{\l@dcsnotetext@l}{#1}%
6558   \else%
6559     \listgadd{\l@dcsnotetext}{#1}%
6560   \fi
6561 }
6562 \newcommand*{\vl@drsnote}[1]{%
6563   \ifledRcol{%
6564     \@l@tempcntb=\sidenote@marginR%
6565     \ifnum\@l@tempcntb>\@ne%
6566       \advance\@l@tempcntb by\page@numR%
6567     \fi%
6568   \else%
6569     \@l@tempcntb=\sidenote@margin%
6570     \ifnum\@l@tempcntb>\@ne%
6571       \advance\@l@tempcntb by\page@num%
6572     \fi%
6573   \fi%
6574   \ifodd\@l@tempcntb%
6575     \listgadd{\l@dcsnotetext}{#1}%
6576   \else%
6577     \listgadd{\l@dcsnotetext@r}{#1}%
6578   \fi%
6579 }
6580 \newcommand*{\vl@dcsnote}[1]{\listgadd{\l@dcsnotetext}{#1}}
6581
6582 \newcommand{\vl@disnote}[1]{%
6583   \ifledRcol{%
6584     \@tempcnta=\page@numR%
6585   \else%
6586     \@tempcnta=\page@num%
6587   \fi%
6588   \ifodd\@tempcnta% ODD => right page => inner side = left side
6589     \vl@dlsnote{#1}%
6590   \else%

```

```

6591 \vl@drsnote{#1}%
6592 \fi%
6593 }%
6594
6595 \newcommand{\vl@dosnote}[1]{%
6596 \ifledRcol%
6597 \@tempcnta=\page@numR%
6598 \else%
6599 \@tempcnta=\page@num%
6600 \fi%
6601 \ifodd\@tempcnta% ODD => right page => outer side = right side
6602 \vl@drsnote{#1}%
6603 \else%
6604 \vl@dlsnote{#1}%
6605 \fi%
6606 }%
6607
6608 %

```

`\setl@dlp@rbox` `\setl@dlprbox{<lednums>}{<tag>}{<text>}` puts `<text>` into the `\l@dlp@rbox` box.  
`\setl@drpr@rbox` And similarly for the right side box. It is these boxes that finally get displayed in the margins.

```

6609 \newcommand*{\setl@dlp@rbox}[1]{%
6610 \begingroup%
6611 \parindent\z@\hsize=\ledlsnotewidth%
6612 \ledlsnotefontsetup%We kept it outside of the vbox, because can affect
the ragging
6613 \global\setbox\l@dlp@rbox%
6614 \ifleftnoteup%
6615 =\vbox to\z@{\ledlsnotefontsetup\vss #1}}%We put \
ledlsnotefontsetup inside footnote because required for color command. Note
the {} to keep setting local.
6616 \else%
6617 =\vbox to 0.70\baselineskip{\ledlsnotefontsetup\strut#1\vss}}%
6618 \fi%
6619 \endgroup%
6620 }
6621
6622 \newcommand*{\setl@drp@rbox}[1]{%
6623 \begingroup%
6624 \parindent\z@\hsize=\ledrsnotewidth%
6625 \ledrsnotefontsetup%We kept it outside of the vbox, because can affect
the ragging
6626 \global\setbox\l@drp@rbox%
6627 \ifrighnoteup%
6628 =\vbox to\z@{\ledrsnotefontsetup\vss#1}}%We put \ledrsnotefontsetup
inside footnote because required for color command. Note the {} to keep
setting local.
6629 \else%

```

```

6630     =\vbox to0.7\baselineskip{{\ledrsnotefontsetup\strut#1\vss}}}%
6631     \fi%
6632     \endgroup%
6633 }%
6634 \newif\ifleftnoteup
6635 \leftnoteuptrue
6636 %

```

**\@sidenotesep** This macro is used to separate sidenotes of the same line.

```

6637 \newcommand{\setsidenotesep}[1]{\gdef\@sidenotesep{#1}}
6638 \newcommand{\@sidenotesep}{, }
6639 %

```

**\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.

```

6640 \newcommand*{\affixside@note}{%
6641     \prepare@edindex@for@note{\the\page@num|\the\line@num|\the\subline@num|\
the\page@num|\the\line@num|\the\subline@num|}%
6642     \def\sidenotecontent@{}%
6643     \numgdef{\itemcount@}{0}%
6644     \def\do##1{%
6645         \ifnumequal{\itemcount@}{0}%
6646             {%
6647                 \appto\sidenotecontent@{##1}}% Not print not separator before
the 1st note
6648                 \appto\sidenotecontent@{\@sidenotesep ##1}%
6649             }%
6650         \numgdef{\itemcount@}{\itemcount@+\@ne}%
6651     }%
6652     \dolistloop{\l@dcnotesetext}%
6653     \ifnumgreater{\itemcount@}{1}{\led@err@ManySidenotes}{}%
6654 %

```

And we do the same for left and right notes (not movable).

```

6655 \gdef\@templ@d{%
6656 \gdef\@templ@n{\l@dcnotesetext\l@dcnotesetext@l\l@dcnotesetext@r}%
6657 \ifx\@templ@d\@templ@n \else%
6658     \if@twocolumn%
6659         \if@firstcolumn%
6660             \setl@dlp@rbox{##1}{\sidenotecontent@}%
6661         \else%
6662             \setl@drp@rbox{\sidenotecontent@}%
6663         \fi%
6664     \else%

```

```

6665 \l@dttempcntb=\sidenote@margin%
6666 \ifnum\l@dttempcntb>\@ne%
6667 \advance\l@dttempcntb by\page@num%
6668 \fi%
6669 \ifodd\l@dttempcntb%
6670 \setl@drp@rbox{\sidenotecontent@}%
6671 \gdef\sidenotecontent@{}%
6672 \numgdef{\itemcount@}{0}%
6673 \dolistloop{\l@dcsnotetext@l}%
6674 \ifnumgreater{\itemcount@}{1}{\led@err@ManyLeftnotes}{}%
6675 \setl@dlp@rbox{\sidenotecontent@}%
6676 \else%
6677 \setl@dlp@rbox{\sidenotecontent@}%
6678 \gdef\sidenotecontent@{}%
6679 \numgdef{\itemcount@}{0}%
6680 \dolistloop{\l@dcsnotetext@r}%
6681 \ifnumgreater{\itemcount@}{1}{\led@err@ManyRightnotes}{}%
6682 \setl@drp@rbox{\sidenotecontent@}%
6683 \fi%
6684 \fi%
6685 \fi%
6686 \advance\@edindex@fornote@m@ne%
6687 }
6688 %

```

## 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 `\@iiminipage` and `\endminipage` macros. We will arrange this so that additional series can be easily added.

`\l@dfetbeginmini` These will be the hooks in `\@iiminipage` and `\endminipage`.  
`\l@dfetendmini` They can be extended to handle other things if necessary.

```

6689 \ifnoledgroup@else%
6690 \newcommand*{\l@dfetbeginmini}{\@ledgrouptrue\l@dedbeginmini\l@dfambeginmini}
6691 \newcommand*{\l@dfetendmini}{%
6692 \IfStrEq{critical-familiar}{\@mpfnpos}%
6693 {\l@dedendmini\l@dfamendmini}%
6694 {%
6695 \IfStrEq{familiar-critical}{\@mpfnpos}%
6696 {\l@dfamendmini\l@dedendmini}%
6697 {\do@feet@custom@order{mp@}{\@mpfnpos}}}%
6698 }%
6699 }%
6700 %

```

`\l@dedbeginmini` These handle the initiation and closure of critical footnotes in a minipage environment.

`\l@dedendmini`

`\mp@append@Xnotes`

```

6701 \newcommand*{\l@dedbeginmini}{%
6702   \unless\ifnocritical%
6703     \def\do##1{%
6704       \csletcs{v##1footnote}{mpv##1footnote}%
6705     }%
6706     \dolistloop{\@series}%
6707   \fi%
6708 }
6709 \newcommand*{\l@dedendmini}{%
6710   \unless\ifnocritical%
6711     \ifl@dpairing%
6712       \ifledRcol%
6713         \flush@notesR%
6714       \else%
6715         \flush@notes%
6716       \fi%
6717     \fi
6718     \def\do##1{%
6719       \mp@append@Xnotes{##1}%
6720     }%
6721     \dolistloop{\@series}%
6722   \fi%
6723 }%
6724 \newcommand{\mp@append@Xnotes}[1]{%
6725   \ifvoid\csuse{mp#1footins}\else%
6726     \ifl@dpairing%
6727       \ifparledgroup%
6728         \ifledRcol%
6729           \dingdef{\parledgroup@beforenotesR}{\parledgroup@beforenotesR+\skip
\@nameuse{mp#1footins}}%
6730         \else%
6731           \dingdef{\parledgroup@beforenotesL}{\parledgroup@beforenotesL+\skip
\@nameuse{mp#1footins}}%
6732         \fi%
6733       \fi%
6734     \fi%
6735     \ifcsstring{series@display#1}{paragraph}{}{%
6736       \setbox\@nameuse{mp#1footins}=\vbox{%
6737         \csuse{Xnotefontsize@#1}%
6738         \ifcsdef{Xhsize}\csuse{series@display#1}@#1{%
6739           \hsize\csuse{Xhsize}\csuse{series@display#1}@#1}%
6740         }{%
6741           \noindent\csuse{Xtxtbeforenotes@#1}%
6742           \unvbox\@nameuse{mp#1footins}%
6743           \@parboxrestore%
6744         }%
6745       }%
6746       \csuse{mp#1footgroup}{#1}%

```

```

6747 \fi%
6748 }%
6749 %

```

`\l@dfambeginmini` These handle the initiation and closure of familiar footnotes in a minipage environment.

```

\l@dfamendmini
\mp@append@notesX
6750 \newcommand*\l@dfambeginmini{%
6751 \unless\ifnofamiliar%
6752 \def\do##1{\csletcs{vfootnote##1}{mpvfootnote##1}}%
6753 \dolistloop{\@series}%
6754 \fi%
6755 }%
6756
6757 \newcommand*\l@dfamendmini{%
6758 \unless\ifnofamiliar%
6759 \def\do##1{%
6760 \mp@append@notesX{##1}%
6761 }%
6762 \dolistloop{\@series}%
6763 \fi%
6764 }%
6765 \newcommand{\mp@append@notesX}[1]{%
6766 \ifvoid\csuse{mpfootins#1}\else%
6767 \csuse{mpfootgroup#1}{#1}%
6768 \fi%
6769 }%
6770 %

```

`\@iiminipage` This is our extended form of the kernel `\@iiminipage` defined in `ltboxes.dtx`.

```

6771 \patchcmd%
6772 {\@iiminipage}%
6773 {\let\@footnotetext\@mpfootnotetext}%
6774 {\let\@footnotetext\@mpfootnotetext\l@dfeetbeginmini}%
6775 {}%
6776 {\led@error@fail@patch@\@iiminipage}%
6777 %

```

`\endminipage` This is our extended form of the kernel `\endminipage` defined in `ltboxes.dtx`.

```

6778 \patchcmd%
6779 {\endminipage}%
6780 {\footnoterule}%
6781 {\footnoterule\l@advance@parledgroup@beforenormalnotes}%
6782 {}%
6783 {\led@error@fail@patch@endminipage}%
6784
6785 \patchcmd%
6786 {\endminipage}%
6787 {\@minipagefalse}%

```

```

6788 {\l@dfeetendmini\@minipagefalse}%
6789 {}%
6790 {\led@error@fail@patch@endminipage}
6791
6792 %

```

`\l@dunboxmpfoot` `\@ldunboxmpfoot` insert normal footnotes for `ledgroup`.

`edgroup@beforenormalnotes`

```

6793 \newcommand*{\l@dunboxmpfoot}{%
6794   \vskip\skip\@mpfootins
6795   \normalcolor
6796   \footnoterule
6797   \l@advance@parledgroup@beforenormalnotes
6798   \unvbox\@mpfootins%
6799 }
6800 %

```

When using parallel `ledgroup`, we need to store the vertical space added before footnote, in order to compensate them between left and right pages.

```

6801 \newcommand{\l@advance@parledgroup@beforenormalnotes}{%
6802   \ifparledgroup
6803     \ifl@dpairing
6804       \ifledRcol
6805         \dimdef{\parledgroup@beforenotesR}{\parledgroup@beforenotesR+
skip\@mpfootins}
6806       \else
6807         \dimdef{\parledgroup@beforenotesL}{\parledgroup@beforenotesL+
skip\@mpfootins}
6808       \fi
6809     \fi
6810   \fi
6811 }
6812 %

```

`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`.

```

6813
6814 \newenvironment{ledgroup}{%
6815   \resetprevpage@num%
6816   \def\@mpfn{mpfootnote}\def\thempfn{\thempfootnote}\c@mpfootnote\z@%
6817   \let\@footnotetext\@mpfootnotetext
6818   \l@dfeetbeginmini%
6819 }{%
6820   \par
6821   \unskip
6822   \ifvoid\@mpfootins\else
6823     \l@dunboxmpfoot

```

```

6824 \fi
6825 \l@dfeetendmini%
6826 \@ledgroupfalse%
6827 }
6828
6829
6830 %

```

```

\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.

```

6831 \newenvironment{ledgroupsize}[2][1]{%
6832 %

```

Set the various text measures.

```

6833 \hsize #2\relax
6834 %

```

Initialize fills for centering.

```

6835 \let\ledllfill\hfil
6836 \let\ledrlfill\hfil
6837 \def\@tempa{#1}\def\@tempb{1}%
6838 %

```

Left adjusted numbered lines

```

6839 \ifx\@tempa\@tempb
6840 \let\ledllfill\relax
6841 \else
6842 \def\@tempb{r}%
6843 \ifx\@tempa\@tempb
6844 %

```

Right adjusted numbered lines

```

6845 \let\ledrlfill\relax
6846 \fi
6847 \fi
6848 %

```

Set up the footnoting.

```

6849 \def\@mpfn{mpfootnote}\def\thempfn{\thempfootnote}\c@mpfootnote\z@
6850 \let\@footnotetext\@mpfootnotetext
6851 \l@dfeetbeginmini%
6852 }{%
6853 \par
6854 \unskip
6855 \ifvoid\@mpfootins\else
6856 \l@dunboxmpfoot
6857 \fi

```



```

6858 \l@dfetendmini%
6859 }
6860
6861 %

```

Close the \ifnoledgroup@\else.

```

6862 \fi%
6863 %

```

`\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`.

```

6864 \newif\ifledgroupnotesL@
6865 \newif\ifledgroupnotesR@
6866 %

```

## XXVI Indexing

Here is some code for indexing using page and line numbers.

### XXVI.1 Looking on package order

First, ensure that `imakeidx` or `indextools` is loaded *before* `eledmac`.

```

6867 \AtBeginDocument{%
6868 \unless\ifl@imakeidx%
6869 \@ifpackageloaded{imakeidx}{\led@error@PackageAfterEledmac{imakeidx}}{}
6870 %
6871 \fi%
6872 \unless\ifl@indextools%
6873 \@ifpackageloaded{indextools}{\led@error@PackageAfterEledmac{indextools}}{}
6874 %
6875 \fi%
6876 \unless\ifl@footmisc%
6877 \@ifpackageloaded{footmisc}{\led@error@PackageAfterEledmac{footmisc}}{}
6878 %
6879 \fi%
6880 }
6881 %

```

### XXVI.2 Auxiliary macros for `\edindex`

`\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
6879 \newcommand{\pagelinesep}{-}
6880 \newcommand{\edindexlab}{${&}}
6881 \newcounter{labidx}

```

```

6882 \setcounter{labidx}{0}
6883
6884 %

```

**\doedindexlabel** This macro sets an \edlabel.

```

6885 \newcommand{\doedindexlabel}{%
6886   \stepcounter{labidx}%
6887   \edlabel{\edindexlab\thelabidx}%
6888 }
6889
6890 %

```

**\thepageline** This macro makes up the page/line number combo from the label/ref. The associated counter is never directly used, but it is required in order to not have any error message with \edgls.

```

6891 \newcounter{pageline}%
6892 \renewcommand{\thepageline}{%
6893   \thepage%
6894   \pagelinesep%
6895   \xlineref{\edindexlab\thelabidx}%
6896 }
6897 %

```

**\thestartpageline** These macros make up the page/line start/end number when the \edindex command is called in critical notes.  
**\theendpageline**

```

6898 \newcommand{\thestartpageline}{%
6899   \l@dparsedstartpage%
6900   \pagelinesep%
6901   \l@dparsedstartline%
6902 }
6903 \newcommand{\theendpageline}{%
6904   \l@dparsedendpage%
6905   \pagelinesep%
6906   \l@dparsedendline%
6907 }
6908 %

```

### XXVI.3 Code specific to \edindex in critical footnotes

**\@edindex@fornote@** This counter is incremented at the beginning of each note (either a footnote or a side-note), and decremented at the end of each note. If its value is greater than 0, that means we are inside a note.

```

6909 \newcount\@edindex@fornote@
6910 %

```

`\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 `|`.

```

6911 \newcommand{\prepare@edindex@fornote}[1]{%
6912   \l@dp@rsefootspec#1|}%
6913   \advance\@edindex@fornote@\@ne%
6914 }
6915 %

```

`\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.

```

6916 \newcommand{\get@edindex@ledinnote@command}{%
6917   \ifxindy%
6918     \gdef\@ledinnote@command{%
6919       ledinnote\thelabidx%
6920     }%
6921     \ifxindyhyperref%
6922       \immediate\write\eledmac@xindy@out{%
6923         (define-attributes ("ledinnote\thelabidx"))^^J
6924         \space\space(markup-locref^^J
6925           \eledmacmarkuplocdepth^^J
6926           :open "\string\ledinnote[\edindexlab\thelabidx]{\@index@command
6927         }{"^^J
6928           :close "}"^^J
6929           :attr "ledinnote\thelabidx"^^J
6930         )
6931       }%
6932     \else%
6933       \immediate\write\eledmac@xindy@out{%
6934         (define-attributes ("ledinnote\thelabidx"))^^J
6935         \space\space(markup-locref^^J
6936           \eledmacmarkuplocdepth^^J
6937           :open "\string\ledinnote{\@index@command}{"}^^J
6938           :close "}"^^J
6939           :attr "ledinnote\thelabidx"^^J
6940         )
6941       }%
6942     \fi%
6943   }%

```

If we do not use `xindy` option, `\@ledinnote@command` will produce something like `ledinnote{formattingcommand}`.

```

6943   \else%
6944     \gdef\@ledinnote@command{%
6945       ledinnote[\edindexlab\thelabidx]{\@index@command}%

```

```

6946 }%
6947 \fi%
6948 }
6949 %

```

## XXVI.4 Analysis of command in indexed text

`\get@index@command` This macro is used to analyze if a text to be indexed has a command after a |.

```

6950 \def\get@index@command#1|#2+{%
6951   \gdef\@index@txt{#1}%
6952   \gdef\@index@command{#2}%
6953   \xdef\@index@parenthesis{%
6954     \IfBeginWith{\@index@command}{(}{%
6955       \StrGobbleLeft{\@index@command}{1}{\@index@command}%
6956       \global\let\@index@command\@index@command@%
6957       \xdef\@index@parenthesis{(%
6958     )}%
6959     \IfBeginWith{\@index@command}{)}{%
6960       \StrGobbleLeft{\@index@command}{1}{\@index@command}%
6961       \global\let\@index@command\@index@command@%
6962       \xdef\@index@parenthesis{)%
6963     }%
6964   }
6965 %

```

## XXVI.5 Code for the formatted index

`\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`

```

6966 \newcommandx{\ledinnote}[3][1,usedefault]{%
6967   \ifboolexpr{%
6968     test{\ifdefequal{\iftrue}{\ifHy@hyperindex}}%
6969     or%
6970     bool {xindyhyperref@}%
6971   }%
6972   {%
6973     \csuse{#2}{\hyperlink{#1}{\ledinnotemark{#3}}}%
6974   }%
6975   {%
6976     \csuse{#2}{\ledinnotemark{#3}}%
6977   }%
6978 }%
6979 \newcommand{\ledinnotehyperpage}[2]{\csuse{#1}{\ledinnotemark{\hyperpage{#2}}}}%
6980 \newcommand{\ledinnotemark}[1]{#1\emph{n}}%
6981 %

```

## XXVI.6 Main code

Eledmac and ledmac were using the specific indexing tools of the memoir in order to allow multiple index. However, eldedmac 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
\dummy@edindex
6982 \newcommandx{\@wredindex}[2][1=\expandonce\jobname,usedefault]{%#1 = the
index name, #2 = the text
6983 \begingroup%
6984 \let\emph\@firstofone%
6985 \let\textbf\@firstofone%
6986 \let\textit\@firstofone%
6987 \let\textmd\@firstofone%
6988 \let\textnormal\@firstofone%
6989 \let\textrm\@firstofone%
6990 \let\textsc\@firstofone%
6991 \let\textsf\@firstofone%
6992 \let\textsl\@firstofone%
6993 \let\texttt\@firstofone%
6994 \let\textup\@firstofone%
6995 \xdef\@tmp{#2}%To be used in IfSubStr instead of #2 directly. Avoid
some expansion bugs (for example with \edindex{textsc{something}})
6996 \endgroup%
6997 \ifl@imakeidx%
6998 \ifnum\@edindex@fornote@>\z@%
6999 \IfSubStr[1]{\@tmp}{|}{\get@index@command#2+}{\get@index@command#2|+}
%
7000 \get@edindex@ledinnote@command%
7001 \expandafter\imki@wrindexentry{#1}{\@index@txt|(\@ledinnote@command
}{\thestartpageline}%
7002 \expandafter\imki@wrindexentry{#1}{\@index@txt|)\@ledinnote@command
}{\theendpageline}%
7003 \else%
7004 \get@edindex@hyperref{#2}%
7005 \imki@wrindexentry{#1}{\@index@txt\@edindex@hyperref}{\thepageline}%
7006 \fi%
7007 \else%
7008 \ifnum\@edindex@fornote@>\z@%
7009 \IfSubStr[1]{\@tmp}{|}{\get@index@command#2+}{\get@index@command#2|+}
%
7010 \get@edindex@ledinnote@command%
7011 \expandafter\protected@write\@indexfile{}%
7012 {\string\indexentry{\@index@txt|(\@ledinnote@command}{\thestartpageline}

```

```

7013 }%
7014   \expandafter\protected@write\@indexfile{}%
7015   {\string\indexentry{\@index@txt|)\@ledinnote@command}{\theendpageline}
7016   }%
7017   \else%
7018     \protected@write\@indexfile{}%
7019     {\string\indexentry{#2}{\thepageline}
7020     }%
7021     \fi%
7022   \fi%
7023   \endgroup
7024   \@esphack%
7025 }
7026 %

```

Need to add the definition of `\edindex` to `\makeindex`, and initialise `\edindex` to do nothing.

```

7027 \pretocmd{\makeindex}{%
7028   \def\edindex{%
7029     \ifboolexpr{bool{numbering} or bool{numberingR} or bool{
7030       l@dprintingpages} or bool{l@dprintingcolumns}}{%
7031       \@bsphack%
7032       \doedindexlabel%
7033       \begingroup%
7034       \@sanitize%
7035       \@wredindex%
7036     }%
7037     {%
7038       \led@warn@edinde@outsidenumbering%
7039       \index%
7040     }%
7041   }%
7042 }%
7043 {\led@error@fail@patch@makeindex}%
7044 \newcommand{\edindex}[1]{\@bsphack\@esphack}
7045 \newcommandx{\dummyedindex}[2][1=\expandonce\jobname,usedefault]{}%
7046 %

```

## XXVI.7 Hyperlink

`\hyperlinkformat` `\hyperlinkformat` command is to be used to have both a internal hyperlink and a format, when indexing.

```

7047 \newcommand{\hyperlinkformat}[3]{%
7048   \ifstrempy{#1}%
7049   {\hyperlink{#2}{#3}}%
7050   {\csuse{#1}{\hyperlink{#2}{#3}}%
7051   }}
7052 %

```

`\hyperlinkR` `\hyperlinkR` command is to be used to create a internal hyperlink and `\ledRflag`, when indexing.

```
7053 \newcommand{\hyperlinkR}[2]{%
7054   \hyperlink{#1}{#2\@Rlineflag}%
7055 }%
7056
7057 %
```

`\hyperlinkformatR` `\hyperlinkformatR` command is to be used to create a internal hyperlink, a format and a `\@Rlineflag`, when indexing.

```
7058 \newcommand{\hyperlinkformatR}[3]{%
7059   \hyperlinkformat{#1}{#2}{#3\@Rlineflag}%
7060 }%
7061
7062 %
```

`\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`).

```
7063 \newcommand{\get@edindex@hyperref}[1]{%
7064   %
```

We have to disable temporary spaces to work through a `xstring` bug (or feature?)

```
7065   \edef\temp@{%
7066     \catcode`\ =9 %space need for catcode
7067     \detokenize{#1}%For active character in unicode
7068     \catcode`\ =10 % space need for catcode
7069   }%
7070 %
```

Now, we define `\@edindex@hyperref` if the hyperindex of `hyperref` is enabled.

```
7071 \ifdefequal{\iftrue}{\ifHy@hyperindex}{%
7072   \IfSubStr{\temp@}{|}%
7073   {\get@index@command#1+%
7074     \ifledRcol%
7075       \gdef\@edindex@hyperref{|\@index@parenthesis %space kept
7076         hyperlinkformatR{\@index@command}%
7077         {\edindexlab\thelabidx}}%
7078     \else%
7079       \gdef\@edindex@hyperref{|\@index@parenthesis %space kept
7080         hyperlinkformat{\@index@command}%
7081         {\edindexlab\thelabidx}}%
7082     \fi%
7083   }%
7084   {\get@index@command#1+%
7085     \ifledRcol%
7086       \gdef\@edindex@hyperref{\hyperlinkR{\edindexlab\thelabidx}}%
7087     \else%
```

```

7088 \gdef\@edindex@hyperref{|hyperlink{\edindexlab\thelabidx}}%
7089 \fi%
7090 }%
7091 }%
7092 %

7093 % If we use both xindy and hyperref, first get the \protect\cs{
index@command} command.
7094 % Then define \protect\cs{@edindex@hyperref} in the form \verb+eledmacXXX+
7095 % \begin{macrocode}
7096 {\ifxindyhyperref%
7097 \IfSubStr{\temp@}{|}%
7098 {\get@index@command#1+}%
7099 {\get@index@command#1|+}%
7100 \gdef\@edindex@hyperref{|eledmac\thelabidx}%
7101 %

```

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`.

```

7102 \IfStrEq{\@index@parenthesis}{(}%
7103 {%
7104 \csxdef{xindyparenthesis@\@index@txt}{\thelabidx}%
7105 \gdef\@edindex@hyperref{| (eledmac\thelabidx}%
7106 }%
7107 {}%
7108 %

```

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`.

```

7109 \IfStrEq{\@index@parenthesis}{)}%
7110 {%
7111 \xdef\@edindex@hyperref{)|eledmac\csuse{xindyparenthesis@\@index@txt}}%
7112 \global\csundef{xindyparenthesis@\@index@txt}%
7113 }%
7114 %

```

Write in the `.xdy` file the attributes of the location.

```

7115 {%
7116 \immediate\write\eledmac@xindy@out{%
7117 (define-attributes ("eledmac\thelabidx"))^^J
7118 \space\space(markup-locoref^^J
7119 \eledmacmarkuplocorefdepth^^J
7120 :open "\string\hyperlink%
7121 \ifledRcol R\fi%
7122 {\edindexlab\thelabidx}%
7123 {\ifdefempty{\@index@command}}%

```



```

7124         {}%
7125         {\@backslashchar\@index@command}%
7126         {"^^J
7127         :close "}"^^J
7128         :attr "eledmac\thelabidx"^^J
7129         )
7130     }%
7131 }%
7132 %

```

And now, in any other case.

```

7133 \else%
7134 \gdef\@index@txt{#1}%
7135 \gdef\@edindex@hyperref{}%
7136 \fi%
7137 }%
7138 }
7139 %

```

## XXVI.8 ‘innote’ and ‘notenumber’ option of indextols package

**\led@set@index@fornote** The `\led@set@index@fornote` is called when a familiar footnote is inserted — and not when it is read — and changes the `\index` command depending of the option of the `indextools` package. Its only argument is the note series.

```

7140 \newcommand{\led@set@index@fornote}[1]{%
7141 \ifbool{indtl@innote}%
7142 {\let\index\nindex}%
7143 {}%
7144 \ifbool{indtl@notenumber}%
7145 {%
7146 \renewcommand{\index}[2][\indtl@jobname]{%
7147 \orig@@index[##1]{%
7148 ##2|innotenumber{\this@footnoteX@reading}%
7149 }%
7150 }%
7151 }%
7152 {}%
7153 }%
7154 %

```

**\led@reinit@index@fornote** The `\led@reinit@index@fornote` just reset the default value of `\index`.

```

7155 \newcommand{\led@reinit@index@fornote}{%
7156 \ifbool{indtl@innote}%
7157 {\let\index\orig@@index}%
7158 {}%
7159 \ifbool{indtl@notenumber}%

```

```

7160 {\let\index\orig@index}%
7161 {}%
7162 }%
7163 %

```

## XXVII Glossaries

Here, we define the \gls-like commands prefixed by ed, only if the package glossaries is loaded.

```

7164 \AtBeginDocument{%
7165   \ifpackageloaded{glossaries}{%
7166     %

```

First those which arguments are [*options*]{*label*}[*insert*].

```

7167   \renewcommand{\do}[1]{%
7168     \expandafter\DeclareRobustCommand\csname ed#1\endcsname[3][1,3,
usedefault]{%
7169       \doedindexlabel%
7170       \csname#1\endcsname[counter=pageline,##1]{##2}[##3]%
7171     }%
7172     \expandafter\WithSuffix\expandafter\DeclareRobustCommand\csname ed
#1\endcsname*[3][1,3,usedefault]{%
7173       \doedindexlabel%
7174       \csname#1\endcsname*[counter=pageline,##1]{##2}[##3]%
7175     }%
7176   }%
7177   \docsvlist{%
7178     gls,%
7179     Gls,%
7180     GLS,%
7181     glspl,%
7182     Glspl,%
7183     GLSpl,%
7184     glstext,%
7185     Glstext,%
7186     GLStext,%
7187     Glsfirst,%
7188     GLSfirst,%
7189     glsplural,%
7190     Glsplural,%
7191     GLSplural,%
7192     glsfirstplural,%
7193     Glsfirstplural,%
7194     GLSfirstplural,%
7195     glsname,%
7196     Glsname,%
7197     GLSname,%
7198     glssymbol,%

```

```

7199      Glssymbol,%
7200      GLSsymbol,%
7201      glsdsc,%
7202      GLsdsc,%
7203      GLSdsc,%
7204      glsuseri,%
7205      Glsuseri,%
7206      GLSuseri,%
7207      glsuserii,%
7208      Glsuserii,%
7209      GLSuserii,%
7210      glsuseriii,%
7211      Glsuseriii,%
7212      GLSuseriii,%
7213      glsuseriv,%
7214      Glsuseriv,%
7215      GLSuseriv,%
7216      glsuserv,%
7217      Glsuserv,%
7218      GLSuserv,%
7219      glsuservi,%
7220      Glsuservi,%
7221      GLSuservi%
7222      }%
7223 %

```

Then those which arguments are [*options*]{*label*}{*link text*}.

```

7224      \renewcommand{\do}[1]{%
7225      \expandafter\DeclareRobustCommandx\csname ed#1\endcsname[3][1,
usedefault]{%
7226      \doedindexlabel%
7227      \csname#1\endcsname[counter=pageline,##1]{##2}{##3}%
7228      }%
7229      \expandafter\WithSuffix\expandafter\DeclareRobustCommandx\csname ed
#1\endcsname*[3][1,usedefault]{%
7230      \doedindexlabel%
7231      \csname#1\endcsname*[counter=pageline,##1]{##2}{##3}%
7232      }%
7233      }%
7234      \docsvlist{glsdisp,glslink}%
7235 %

```

Then those which arguments are [*options*]{*label*}.

```

7236      \renewcommand{\do}[1]{%
7237      \expandafter\DeclareRobustCommandx\csname ed#1\endcsname[2][1,
usedefault]{%
7238      \doedindexlabel%
7239      \csname#1\endcsname[counter=pageline,##1]{##2}%
7240      }%

```

```

7241 \expandafter\WithSuffix\expandafter\DeclareRobustCommand\csname ed
#1\endcsname*[2][1,usedefault]{%
7242 \doedindexlabel%
7243 \csname#1\endcsname*[counter=pageline,##1]{##2}%
7244 }%
7245 }%
7246 \docsvlist{glsadd}%
7247 }{}%
7248 }%
7249 %

```

## XXVIII 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`.

### XXVIII.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`.

`\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.

```

7250 \def\@hangingsymbol{}
7251 \newcommand*\sethangingsymbol[1]{%
7252 \gdef\@hangingsymbol{#1}%
7253 }%
7254 \newif\ifinstanza
7255 %

```

`\inserthangingsymbol` The boolean `\ifinserthangingsymbol` is set to TRUE when `\@lock` is greater than 1, i.e. when we are not in the first line of a verse. The switch of `\ifinserthangingsymbol` is made in `\do@line` before the printing of line but after the line number calculation.

```

7256 \newif\ifinserthangingsymbol
7257 \newcommand\inserthangingsymbol{%
7258 \ifinserthangingsymbol%
7259 \ifinstanza%
7260 \@hangingsymbol%
7261 \fi%
7262 \fi%
7263 }
7264 %

```

## XXVIII.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.

```
7265 \newcommand*\ampersand{\char`\&}
7266
7267 %
```

## XXVIII.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.

```
7268 \chardef\body=\catcode`\@
7269 \catcode`\@=11
7270 \chardef\next=\catcode`\&
7271 \catcode`\&=\active
7272
7273 %
```

## XXVIII.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.

```
7274 \newcount\stanza@count
7275 \newlength{\stanzaindentbase}
7276 \setlength{\stanzaindentbase}{20pt}
7277
7278 %
```

**\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.

```
7279 \def\strip@szacnt#1,#2|{\def\@tempb{#1}\def\@tempa{#2|}}
7280 \newcommand*\setstanzavalues[2]{\def\@tempa{#2,,|}%
7281   \stanza@count\z@
7282   \def\next{\expandafter\strip@szacnt\@tempa
7283     \ifx\@tempb\empty\let\next\relax\else
7284     \expandafter\mathchardef\csname #1@number\stanza@count
7285     \@endcsname\@tempb\relax
7286     \advance\stanza@count\@ne\fi\next}%
```

```

7287 \next}
7288
7289 %

```

**\setstanzaindents** In the original edmac, `\setstanzavalues{sza}{⟨...⟩}` had to be called to set the indents, and similarly `\setstanzavalues{szp}{⟨...⟩}` 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).

```

7290 \newcommand*{\setstanzaindents}[1]{\setstanzavalues{sza}{#1}}
7291 \newcommand*{\setstanzapenalties}[1]{\setstanzavalues{szp}{#1}}
7292 %
7293 %

```

**\managestanza@modulo** Since version 0.13, the `stanzaindentsrepetition` counter can be used when the indentation is repeated every *n* verses. The `\managestanza@modulo` is a command which modifies the counter `stanza@modulo`. The command adds 1 to `stanza@modulo`, but if `stanza@modulo` is equal to the `stanzaindentsrepetition` counter, the command restarts it.

```

7294 \newcounter{stanzaindentsrepetition}
7295 \newcount\stanza@modulo
7296
7297 \newcommand*{\managestanza@modulo}[0]{%
7298   \advance\stanza@modulo\@ne%
7299   \ifnum\stanza@modulo>\value{stanzaindentsrepetition}%
7300     \stanza@modulo\@ne%
7301   \fi%
7302 }
7303 %

```

**\stanzaindent** The macro `\stanzaindent`, when called at the beginning of a verse, changes the indentation normally defined for this verse by `\setstanzaindent`. The starred version **\stanzaindent\*** skips the current verse for the repetition of stanza indent.

```

7304 \newcommand{\stanzaindent}[1]{%
7305   \hspace{\dimexpr#1\stanzaindentbase-\parindent\relax}%
7306   \ignorespaces%
7307 }%
7308 \WithSuffix\newcommand\stanzaindent*[1]{%
7309   \stanzaindent{#1}%
7310   \global\advance\stanza@modulo-\@ne%
7311   \ifnum\stanza@modulo=0%
7312     \global\stanza@modulo=\value{stanzaindentsrepetition}%
7313   \fi%
7314   \ignorespaces%
7315 }%
7316 %

```

## XXVIII.5 Numbering stanza

Here, macro for numbering stanza. First, the stanza counter.

```
\thestanza 17 \newcounter{stanza}
7318 \renewcommand{\thestanza}{%
7319 \textbf{\arabic{stanza}}%
7320 }
7321 %
```

`\ifnumberstanza` Then, macro to activate automatically numbering of stanza.

```
7322 \newif\ifnumberstanza%
7323 %
```

`\@insertstanzanumber` Now, macro called at the first line of of verse of a stanza.

```
7324 \newcommand{\@insertstanzanumber}[0]{%
7325 \ifnumberstanza%
7326 \ifl@pairing%
7327 \ifledRcol%
7328 \stanzanumwrapper{\thestanzaR}%
7329 \else%
7330 \stanzanumwrapper{\thestanzaL}%
7331 \fi%
7332 \else%
7333 \stanzanumwrapper{\thestanza}%
7334 \fi%
7335 \setline{1}%
7336 \fi%
7337 }%
7338 %
```

`\@advancestanzanumber` Also a command to advance the counter of stanza.

```
7339 \newcommand{\@advancestanzanumber}[0]{%
7340 \ifnumberstanza%
7341 \ifl@pairing%
7342 \ifledRcol%
7343 \addtocounter{stanzaR}{1}%
7344 \else%
7345 \addtocounter{stanzaL}{1}%
7346 \fi%
7347 \else%
7348 \addtocounter{stanza}{1}%
7349 \fi%
7350 \fi%
7351 }%
7352 %
```

`\stanzanumwrapper` And finally, the wrapper for stanza number

```

7353 \newcommand{\stanzanumwrapper}[1]{%
7354   \flagstanza{#1}%
7355 }%
7356 %

```

## XXVIII.6 Stanza number in note

Here, the command called when printing stanza number in notes.

```

7357 \newcommand{\printstanza}[0]{%
7358   \ifboolexpr{bool{!@dpairing} or bool{!@dprintingpages} or bool{
7359     !@dprintingcolumns}}{%
7360     \ifledRcol{%
7361       \thestanzaR%
7362     }%
7363   }%
7364   \ifledLcol{%
7365     \thestanzaL%
7366   }%
7367 }%
7368 %

```

## XXVIII.7 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. `\sza@penalty`

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.

```

7369 \newcommandx{\stanza@line}[2][1,2,usedefault]{%
7370   \ifnum\value{stanzaindentrepetition}=0
7371   \ifcsdef{sza@\number\stanza@count @}{%
7372     {%
7373       \parindent=\csname sza@\number\stanza@count @\endcsname\
7374       stanzaindentbase%
7375     }%
7376     \led@err@StanzaIndentNotDefined%
7377   }%
7378   \else
7379     \ifcsdef{sza@\number\stanza@modulo @}{%
7380       \parindent=\csname sza@\number\stanza@modulo @\endcsname\
7381       stanzaindentbase%

```



```

7380         \managestanza@modulo%
7381     }%
7382     {%
7383     \led@err@StanzaIndentNotDefined%
7384     }%
7385     \fi
7386     \pstart[#1][#2]\stanza@hang\ignorespaces%
7387 }%
7388 \xdef\stanza@hang{\noexpand\leavevmode\noexpand\startlock
7389     \hangindent\expandafter
7390     \noexpand\csname sz@0@\endcsname\stanzaindentbase
7391     \hangafter\@ne}
7392 \def\sz@penalty{\count@\csname szp@\number\stanza@count @\endcsname
7393     \ifnum\count@>\@M\advance\count@-\@M\penalty-\else
7394     \penalty\fi\count@}
7395 %

```

\@startstanza    Now we have the components of the \stanza macro, which appears at the start of a  
                   \stanza    group of lines. This macro initializes the count and checks to see if hanging indentation  
                   \@stopstanza    and penalties are to be included. Hanging indentation suspends the line count, so that  
 \@EveryStopStanza    the enumeration is by verse line rather than by print line. If the print line count is  
                   \AtEveryStanza    desired, invoke \let\startlock\relax and do the same for \endlock. Here and  
 \AtStartEveryStanza    above we have used \xdef to make the stored macros take up a bit less space, but it also  
 \BeforeEveryStopStanza    makes them more obscure to the reader. Lines of the stanza are delimited by ampersands  
                   \newverse    &. The last line of the stanza must end with \&.

```

7396 \xdef\@startstanza[#1][#2]{%
7397     \noexpand\instanzatrue\expandafter
7398     \begingroup%
7399     \catcode`\noexpand\&\active%
7400     \global\stanza@count\@ne\stanza@modulo\@ne
7401     \noexpand\ifnum\expandafter\noexpand
7402     \csname sz@0@\endcsname=\z@\let\noexpand\stanza@hang\relax
7403     \let\noexpand\endlock\relax\noexpand\else\interlinepenalty
7404     \@M\rightskip\z@ plus 1fil\relax\noexpand\fi\noexpand\ifnum
7405     \expandafter\noexpand\csname szp@0@\endcsname=\z@
7406     \let\noexpand\sz@penalty\relax\noexpand\fi%
7407     \def\noexpand&{%
7408         \noexpand\newverse[] []}%
7409     \def\noexpand\&{\noexpand\@stopstanza}%
7410     \noexpand\@advancestanzanumber%
7411     \noexpand\stanza@line[#1][#2]%
7412     \noexpand\@insertstanzanumber%
7413     \let\par\relax\ignorespaces%No paragraph in verses
7414 }
7415
7416 \newcommandx{\stanza}[2][1,2,usedefault]{%
7417     \ifboolexpr{%
7418         not test{\ifdefvoid{\at@everystanza}}%

```

```

7419 and test{\ifstrempy{#1}}%
7420 and test{\ifstrempy{#2}}}%
7421 {\@startstanza[] [\at@every@stanza] \@start@every@stanza}%
7422 {\@startstanza[#1] [#2] \@start@every@stanza}%
7423 }%
7424
7425 \newcommandx{\@stopstanza}[2][1,2,usedefault]{%
7426   \unskip%
7427   \endlock%
7428   \ifboolexpr{%
7429     not test{\ifdefvoid{\at@every@stop@stanza}}%
7430     and test{\ifstrempy{#1}}%
7431     and test{\ifstrempy{#2}}}%
7432     {\before@every@stop@stanza\pend[] [\at@every@stop@stanza]}%
7433     {\before@every@stop@stanza\pend[#1] [#2]}%
7434   \endgroup%
7435   \instanzafalse%
7436 }
7437
7438 \newcommand{\AtEveryStopStanza}[1]{%
7439   \ifstrempy{#1}%
7440     {\gdef\at@every@stop@stanza{}}%
7441     {\gdef\at@every@stop@stanza{\noindent#1}}%
7442 }%
7443 \WithSuffix\newcommand\AtEveryStopStanza*[1]{%
7444   \ifstrempy{#1}%
7445     {\gdef\at@every@stop@stanza{}}%
7446     {\gdef\at@every@stop@stanza{#1}}%
7447 }%
7448 \def\at@every@stop@stanza{}%
7449
7450 \newcommand{\AtEveryStanza}[1]{%
7451   \ifstrempy{#1}%
7452     {\gdef\at@every@stanza{}}%
7453     {\gdef\at@every@stanza{\noindent#1}}%
7454 }%
7455 \WithSuffix\newcommand\AtEveryStanza*[1]{%
7456   \ifstrempy{#1}%
7457     {\gdef\at@every@stanza{}}%
7458     {\gdef\at@every@stanza{#1}}%
7459 }%
7460
7461
7462
7463 \newcommand{\AtStartEveryStanza}[1]{%
7464   \ifstrempy{#1}%
7465     {\gdef\at@start@every@stanza{}}%
7466     {\gdef\at@start@every@stanza{#1}}%
7467 }%
7468 \def\at@start@every@stanza{}%

```

```

7469 \newcommand{\BeforeEveryStopStanza}[1]{%
7470   \ifstrempy{#1}%
7471     {\gdef\before@every@stop@stanza{}}%
7472     {\gdef\before@every@stop@stanza{#1}}%
7473   }%
7474 \def\before@every@stop@stanza{}%
7475
7476 \newcommandx*\newverse}[4][1,2,3,4,usedefault]{%
7477   \unskip%
7478   \endlock\pend[#1][#3]\sza@penalty\global%
7479   \advance\stanza@count\@ne\stanza@line[#2][#4]%
7480   }
7481
7482 %
7483 %

```

**\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`.

```

7484 \newcommand*\flagstanza}[2][\stanzaindentbase]{%
7485   \hskip -#1\llap{#2}\hskip #1\ignorespaces}
7486
7487 %

```

## XXVIII.8 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.

```

7488 \catcode`\&=\next
7489 \catcode`\@=\body
7490 \setstanzavalues{szp}{0}
7491
7492 %

```

## XXIX Apparatus of Manuscripts

### XXIX.1 User level macro

**\msdata** The user level `\msdata` command only writes the manuscripts data in numbered auxiliary file. There are two associated etoolbox counters.

```

\msdata@cR
7493 \def\msdata@c{}%
7494 \def\msdata@cR{}%

```

```

7495 \newcommand{\msdata}[1]{%
7496   \leavevmode%
7497   \unless\ifstopmsdata@inserted%
7498     \stopmsdata%
7499     \led@warning@msdatawithoutstop%
7500   \fi%
7501   \global\stopmsdata@inserted@false%
7502   \unless\ifledRcol%
7503     \numgdef{\msdata@c}{\msdata@c+1}%
7504     \ifdef{\hypertarget}{%
7505       \edlabel{\msdata@c:start:msdata}%
7506     }{}%
7507     \protected@write\linenum@out{}{%
7508       \string\@msd{#1}%
7509     }%
7510   \else%
7511     \numgdef{\msdata@cR}{\msdata@cR+1}%
7512     \ifdef{\hypertarget}{%
7513       \edlabel{\msdata@cR:start:msdata}%
7514     }{}%
7515     \protected@write\linenum@outR{}{%
7516       \string\@msd{#1}%
7517     }%
7518   \fi%
7519 }%
7520 %

```

**\stopmsdata** The user level `\stopmsdata` command only writes information about the end of manuscripts data in numbered auxiliary file.

```

7521 \newcommand{\stopmsdata}[0]{%
7522   \leavevmode%
7523   \unless\ifledRcol%
7524     \protected@write\linenum@out{}{%
7525       \string\@stopmsd%
7526     }%
7527     \ifdef{\hypertarget}{%
7528       \edlabel{\msdata@c:end:msdata}%
7529     }{}%
7530   \else%
7531     \protected@write\linenum@outR{}{%
7532       \string\@stopmsd%
7533     }%
7534     \ifdef{\hypertarget}{%
7535       \edlabel{\msdata@cR:end:msdata}%
7536     }{}%
7537   \fi%
7538   \global\stopmsdata@inserted@true%
7539 }%
7540 %

```

**\ifstopmsdata@inserted@** The `\ifstopmsdata@inserted@` boolean is set to TRUE at every `\stopmsdata` and reset to FALSE at all `\msdata`. It also set to TRUE at every `\beginnumbering`. It is used to automatically insert `\stopmsdata` if forgotten before `\msdata`

```
7541 \newif\ifstopmsdata@inserted@%
7542 %
```

## XXIX.2 Setting macro

Setting macros for the manuscripts apparatus tools is very easy: they just save their argument in an internal macro.

**\setmsdataseries** In which series of notes will be printed the apparatus of manuscripts?

```
7543 \newcommand{\setmsdataseries}[1]{%
7544   \gdef\@msdata@series{#1}%
7545 }%
7546 \def\@msdata@series{A}%
7547 %
```

**\setmsdataposition** The label for the manuscripts data.

```
7548 \def\ms@data@position{msdata-regular}%
7549 \newcommand{\setmsdataposition}[1]{%
7550   \gdef\ms@data@position{#1}%
7551 }%
7552 %
```

**\setmsdatalabel** The label for the manuscripts data.

```
7553 \def\ms@data@label{Ms.}%
7554 \newcommand{\setmsdatalabel}[1]{%
7555   \gdef\ms@data@label{#1}%
7556 }%
7557 %
```

## XXIX.3 Counters and lists

**\@msd@c** `\@msd@c` is a counter incremented at each `\@msd` read in auxiliary file.

```
7558 \numdef{\@msd@c}{0}
7559 \numdef{\@msd@cR}{0}
7560 %
```

**\add@msd@** `\add@msd@` is a counter incremented at each `\add@msddata`, that is at each time we prepare the insertion of manuscripts data footnote.

```
7561 \numdef{\add@msd@c}{0}%
7562 \numdef{\add@msd@cR}{0}%
7563 %
```

`\@msdata@list` The `\@msdata@list` will contain, for each line, the lists of command to be executed to insert the manuscripts apparatus. It will be filled on `\add@msdata` and looped on `\insert@msdata`, then emptied.

```
7564 \def\@msdata@list{}%
7565 %
```

## XXIX.4 Auxiliary file macros

`\@msd` The `\@msd` macro is written in the auxiliary file. It just defines three macros by `\msdata` macro, which allow us to know the manuscripts data, the line number and the absolute line number where it was called

It also stores the action code 1010 in the list of actions by line.

```
7566 \newcommand{\@msd}[1]{%
7567   \unless\ifledRcol%
7568     \global\newdef{\@msd@c}{\@msd@c+\@ne}%
7569     \csgdef{\msdata@\@msd@c @data}{#1}%
7570     \csxdef{\msdata@\@msd@c @linenumber}{\the\line@num}%
7571     \csxdef{\msdata@\@msd@c @abslinenumber}{\the\absline@num}%
7572     \xright@appenditem{\the\absline@num}\to\actionlines@list%
7573     \xright@appenditem{-1010}\to\actions@list%
7574   \else%
7575     \global\newdef{\@msd@cR}{\@msd@cR+\@ne}%
7576     \csgdef{\msdata@\@msd@cR @dataR}{#1}%
7577     \csxdef{\msdata@\@msd@cR @linenumberR}{\the\line@numR}%
7578     \csxdef{\msdata@\@msd@cR @abslinenumberR}{\the\absline@numR}%
7579     \xright@appenditem{\the\absline@numR}\to\actionlines@listR%
7580     \xright@appenditem{-1010}\to\actions@listR%
7581   \fi%
7582 }%
7583 %
```

`\@stopmsd` Inserted in the auxiliary file by `\stopmsd`, the `\@stopmsd` macro will store in two commands the line number and the absolute line number on which it is called.

```
7584 \newcommand{\@stopmsd}[0]{%
7585   \unless\ifledRcol%
7586     \ifcsundef{\msdata@\@msd@c @stoplinenumber}{%
7587       \csxdef{\msdata@\@msd@c @stopabslinenumber}{\the\absline@num}%
7588       \csxdef{\msdata@\@msd@c @stoplinenumber}{\the\line@num}%
7589     }{}%
7590   \else%
7591     \ifcsundef{\msdata@\@msd@cR @stoplinenumberR}{%
7592       \csxdef{\msdata@\@msd@cR @stopabslinenumberR}{\the\absline@numR}%
7593       \csxdef{\msdata@\@msd@cR @stoplinenumberR}{\the\line@numR}%
7594     }{}%
7595   \fi%
7596 }%
7597 %
```

7598 %

## XXIX.5 Action macro

`\add@msdata` `\add@msdata` is executed on each line when action code 1010 is seen. It will not insert immediately the manuscript data footnote, as action code are executed before the line be typeset, and, consequently, could be on the previous page. So it just stores the manuscript data footnote to `\@msdata@list`.

```

7599 \newcommand{\add@msdata}{%
7600   \bgroup%
7601   \normalfont%
7602   \unless\ifledRcol{%
7603     \numdef{\add@msd@c}{\add@msd@c+\@ne}%
7604     \ifcsdef{@msdata@\add@msd@c @data}{%
7605       \letcs{@data}{@msdata@\add@msd@c @data}%
7606       \edef\l@d@nums{%
7607         000|% Start page = we don't print it
7608         \csuse{@msdata@\add@msd@c @linenumber}|% Start line number
7609         000|% Start subline number, for now, not used
7610         000|% End page number, we don't print it
7611         \ifnumless{\csuse{@msdata@\add@msd@c @stopabslinenumber}}{\csuse{
@lastabsline@forpage@\the\page@num}}}%
7612         {\csuse{@msdata@\add@msd@c @stoplinenumber}}}%End line number if
in the same page
7613         {\csuse{@lastline@forpage@\the\page@num}}}%Otherwiser, last
number of the page
7614         |%
7615         000|% End sub line number, for now, not used
7616         \edfont@info%Font
7617       }%
7618       \@msd@options@fullpagefalse%
7619       \if@firstlineofpage%Try if the data are for the full page. If yes
, will add options to the list.
7620       \unless\if@msdata@insertedfrompreviouspage%
7621       \ifnumless{\csuse{@lastabsline@forpage@\the\page@num}}{\csuse
{@msdata@\add@msd@c @stopabslinenumber}+\@ne}%
7622       {%
7623       \numdef{@tmp}{\add@msd@c+\@ne}%
7624       \ifcsdef{@msdata@\@tmp @abslinenumber}%
7625       {\ifnumequal{\csuse{@msdata@\@tmp @abslinenumber}}{\csuse{
@lastabsline@forpage@\the\page@num}}}%
7626       {}%
7627       {\@msd@options@fullpagetrue}%
7628       }%
7629       {\@msd@options@fullpagetrue}%
7630       }%
7631       {}%
7632       \fi%

```

```

7633 \fi%
7634 \listxadd{\@msdata@list}{%
7635 \@msd@options@iffullpage%
7636 \ifluatex%
7637 \csxdef{footnote@luatextextdir}{\the\textdir}%
7638 \csxdef{footnote@luatexpardir}{\the\pardir}%
7639 \fi%
7640 \csdef{@this@crossref@start}{\add@msd@c:start:msdata}%
7641 \csdef{@this@crossref@end}{\add@msd@c:end:msdata}%
7642 \noexpand\csuse{v\@msdata@series footnote}{\@msdata@series}{\@
expandonce\l@d@nums}{\ms@data@label}{\expandonce\data}}%
7643 \reset@msd@options@iffullpage%
7644 }%
7645 }%
7646 {}%
7647 \else%
7648 \numgdef{\add@msd@cR}{\add@msd@cR+\@ne}%
7649 \ifcsdef{@msdata@\add@msd@cR @dataR}{%
7650 \letcs{\@data}{\@msdata@\add@msd@cR @dataR}%
7651 \edef\l@d@nums{%
7652 000|% Start page = we don't print it
7653 \csuse{@msdata@\add@msd@cR @linenumberR}|% Start line number
7654 000|% Start subline number, for now, not used
7655 000|% End page number, we don't print it
7656 \ifnumless{\csuse{@msdata@\add@msd@cR @stopabslinenumberR}}{\@
csuse{@lastline@forpageR@\the\page@numR}}%
7657 {\csuse{@msdata@\add@msd@cR @stoplinenumberR}}%End line number
if in the same page
7658 {\csuse{@lastline@forpageR@\the\page@numR}}%Otherwiser, last
number of the page
7659 |%
7660 000|% End sub line number, for now, not used
7661 \edefont@info%Font
7662 }%
7663 \@msd@options@fullpagefalse%
7664 \if@firstlineofpageR%
7665 \unless@if@msdata@insertedfrompreviouspage%
7666 \ifnumless{\csuse{@lastabsline@forpageR@\the\page@numR}}{\@
csuse{@msdata@\add@msd@c @stopabslinenumberR}+\@ne}%
7667 {%
7668 \numdef{\@tmp}{\add@msd@cR+\@ne}%
7669 \ifcsdef{@msdata@\@tmp @abslinenumberR}%
7670 {\ifnumequal{\csuse{@msdata@\@tmp @abslinenumberR}}{\csuse{
@lastabsline@forpageR@\the\page@numR}}%
7671 {}%
7672 {\@msd@options@fullpagetrue}%
7673 }%
7674 {\@msd@options@fullpagetrue}%
7675 }%
7676 {}%

```



```

7677     \fi%
7678     \fi%
7679     \listxadd{\@msdata@list}{%
7680       \@msd@options@iffullpage%
7681       \ifluatex%
7682         \csxdef{footnote@luatextextdir}{\the\textdir}%
7683         \csxdef{footnote@luatexpardir}{\the\pardir}%
7684       \fi%
7685       \csdef{@this@crossref@start}{\add@msd@cR:start:msdata}%
7686       \csdef{@this@crossref@end}{\add@msd@cR:end:msdata}%
7687       \noexpand\csuse{v\@msdata@series footnote}{\@msdata@series}{\@
expandonce\l@d@nums}{\ms@data@label}{\expandonce\@data}}%
7688       \reset@msd@options@iffullpage%
7689     }%
7690   }%
7691   {}%
7692   \fi%
7693   \egroup%
7694 }%
7695 %

```

[@insertedfrompreviouspage](#) The `\if@msdata@insertedfrompreviouspage` boolean is set to TRUE if `reledmac` automatically inserts data from previous page in the first line of a page.

```

7696 \newif\if@msdata@insertedfrompreviouspage%
7697 %

```

[dd@msdata@firstlineofpage](#) `\add@msdata@firstlineofpage` is called at the first line of every page. It inserts manuscript data which start on one of the previous pages and continue on this page.

```

7698 \newcommand{\add@msdata@firstlineofpage}{%
7699   \bgroup%
7700   \normalfont%
7701   \unless\ifledRcol%
7702     \ifcsdef{@msdata@\add@msd@c @data}{%
7703       \ifnumless{\the\absline@num-\@one}{\csuse{@msdata@\add@msd@c
@stopabslinenumber}}}%
7704     {%
7705       \global\@msdata@insertedfrompreviouspagetrue%
7706       \letcs{@data}{@msdata@\add@msd@c @data}%
7707       \edef\l@d@nums{%
7708         000|% Start page = we don't print it
7709         \numexpr\the\line@num+\@one\relax|% Start line number = first line
of the page. As \add@msdata@firstlineofpage is called before line number
has been incremented, we increment it for printing
7710         000|% Start subline number, for now, not used
7711         000|% End page number, we don't print it
7712         \ifnumless{\csuse{@msdata@\add@msd@c @stopabslinenumber}}{\csuse{
@lastabsline@forpage@\the\page@num}}}%

```

```

7713     {\csuse{@msdata@\add@msd@c @stoplinenumber}}}%End line number if
      in the same page
7714     {\csuse{@lastline@forpage@\the\page@num}}}%Otherwise, last
      number of the page
7715     |%
7716     000|% End sub line number, for now, not used
7717     \edfont@info%Font
7718     }%
7719     \@msd@options@fullpagefalse%
7720     \ifnumless{\csuse{@lastabsline@forpage@\the\page@num}}{\csuse{
@msdata@\add@msd@c @stopabslinenumber}+\@one}}%We will test if the ms data is
      for the full page
7721     {%
7722     \numdef{\@tmp}{\add@msd@c+\@one}%
7723     \ifcsdef{@msdata@\@tmp @abslinenumber}%
7724     {\ifnumequal{\csuse{@msdata@\@tmp @abslinenumber}}{\csuse{
@lastabsline@forpage@\the\page@num}}%
7725     }%
7726     {\@msd@options@fullpagetrue}%
7727     }%
7728     {\@msd@options@fullpagetrue}%
7729     }%
7730     }%
7731     \listxadd{\@msdata@list}{%
7732     \@msd@options@iffullpage%
7733     \ifluatex%
7734     \csxdef{footnote@luatextextdir}{\the\textdir}%
7735     \csxdef{footnote@luatexpardir}{\the\pardir}%
7736     \fi%
7737     \csdef{@this@crossref@start}{\add@msd@c:start:msdata}%
7738     \csdef{@this@crossref@end}{\add@msd@c:end:msdata}%
7739     \noexpand\csuse{v\@msdata@series footnote}{\@msdata@series}{\
expandonce\l@d@nums}{\ms@data@label}{\expandonce\data}}%
7740     \reset@msd@options@iffullpage%
7741     }%
7742     }%
7743     {\global\@msdata@insertedfrompreviouspagefalse}%
7744     }{}%
7745     \else%
7746     \ifcsdef{@msdata@\add@msd@cR @dataR}{%
7747     \ifnumless{\the\absline@numR-\@one}{\csuse{@msdata@\add@msd@cR
@stopabslinenumber}}}%
7748     {%
7749     \global\@msdata@insertedfrompreviouspagetrue%
7750     \letcs{\@data}{@msdata@\add@msd@cR @dataR}%
7751     \edef\l@d@nums{%
7752     000|% Start page = we don't print it
7753     \numexpr\the\line@numR+\@one\relax|}% Start line number = first
      line of the page. As \add@msdata@firstlineofpage is called before line
      number has been incremented, we increment it for printing

```

```

7754      000|% Start subline number, for now, not used
7755      000|% End page number, we don't print it
7756      \ifnumless{\csuse{@msdata@add@msd@cR @stopabslinenumberR}}{\
csuse{@lastline@forpageR@the\page@numR}}%
7757      {\csuse{@msdata@add@msd@cR @stoplinenumberR}}%End line number
if in the same page
7758      {\csuse{@lastline@forpageR@the\page@numR}}%Otherwise, last
number of the page
7759      |%
7760      000|% End sub line number, for now, not used
7761      \edfont@info%Font
7762      }%
7763      \@msd@options@fullpagefalse%
7764      \ifnumless{\csuse{@lastabsline@forpageR@the\page@numR}}{\csuse{
@msdata@add@msd@cR @stopabslinenumberR}+\@ne}%
7765      {%
7766      \numdef{\@tmp}{\add@msd@cR+\@ne}%
7767      \ifcsdef{@msdata@\@tmp @abslinenumberR}%
7768      {\ifnumequal{\csuse{@msdata@\@tmp @abslinenumberR}}{\csuse{
@lastabsline@forpageR@the\page@numR}}}%
7769      }%
7770      {\@msd@options@fullpagetrue}%
7771      }%
7772      {\@msd@options@fullpagetrue}%
7773      }%
7774      {}%
7775      \listxadd{\@msdata@list}{%
7776      \@msd@options@iffullpage%
7777      \ifluatex%
7778      \csxdef{footnote@luatextextdir}{\the\textdir}%
7779      \csxdef{footnote@luatexpardir}{\the\pardir}%
7780      \fi%
7781      \csdef{@this@crossref@start}{\add@msd@cR:start:msdata}%
7782      \csdef{@this@crossref@end}{\add@msd@cR:end:msdata}%
7783      \noexpand\csuse{v@msdata@series footnote}{\@msdata@series}{\
expandonce\l@d@nums}{\ms@data@label}{\expandonce\data}}%
7784      \reset@msd@options@iffullpage%
7785      }%
7786      }%
7787      {\global\@msdata@insertedfrompreviouspagefalse}%
7788      }{}%
7789      \fi%
7790      \egroup%
7791      }%
7792      %

```

## XXIX.6 Inserting footnote

Just before inserting standard insert (familiar and critical footnotes, sidenotes), we call `\insert@msdata` to insert manuscripts data's footnotes.

```
\insert@msdata 93 \newcommand{\insert@msdata}{%
7794 \def\do##1{##1}%
7795 \dolistloop{\@msdata@list}%
7796 \global\let\@msdata@list\relax%
7797 }%
7798 %
```

## XXIX.7 Other

`\@msd@options@iffullpage` `\@msd@options@iffullpage` sets some options if the manuscripts data are for all the page. `\reset@msd@options@iffullpage` resets them after the footnote. `\if@msd@options@fullpage` is switch to true in `add@msdata@firstlineofpage` if these option must be inserted.

```
7799 \newif\if@msd@options@fullpage%
7800 \newcommand{\@msd@options@iffullpage}[0]{%
7801 \if@msd@options@fullpage%
7802 \noexpand\toggletrue{nonum@}%
7803 \ifdefvoid{\ms@data@label}%
7804 {\noexpand\toggletrue{nosep@}}%
7805 }%
7806 \fi%
7807 }%
7808 \newcommand{\reset@msd@options@iffullpage}[0]{%
7809 \noexpand\togglefalse{nonum@}%
7810 \noexpand\togglefalse{nosep@}%
7811 }%
7812 %
```

## XXX Arrays and tables

### XXX.1 Preamble: macro as environment

The following is borrowed, and renamed, from the `amsmath` package. See also the CTT thread ‘`eeq` and `amstex`’, 1995/08/31, started by Keith Reckdahl and ended definitively by David M. Jones.

Several of the `[math]` macros scan their body twice. This means we must collect all text in the body of an environment form before calling the macro.

`\@emptytoks` This is actually defined in the `amsngen` package.

```
7813 \newtoks\@emptytoks
7814
7815 %
```

The rest is from amsmath.

**\l@denbody** A token register to contain the body.

```
7816 \newtoks\l@denbody
7817
7818 %
```

**\addtol@denbody** `\addtol@denbody{arg}` adds arg to the token register `\l@denbody`.

```
7819 \newcommand{\addtol@denbody}[1]{%
7820   \global\l@denbody\expandafter{\the\l@denbody#1}}
7821
7822 %
```

**\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`.

```
7823 \newcommand{\l@dcollect@body}[1]{%
7824   \l@denbody{\expandafter#1\expandafter{\the\l@denbody}}%
7825   \edef\processl@denbody{\the\l@denbody\noexpand\end{@currentenv}}%
7826   \l@denbody\emptytoks \def\l@dbegin@stack{b}%
7827   \begingroup
7828     \expandafter\let\csname\@currentenv\endcsname\l@dcollect@@body
7829     \edef\processl@denbody{\expandafter\noexpand\csname\@currentenv\endcsname}%
7830     \processl@denbody%
7831   }%
7832
7833 %
```

**\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.

```
7834 \def\l@dpush@begins#1\begin#2{%
7835   \ifx\end#2\else b\expandafter\l@dpush@begins\fi}
7836
7837 %
```

**\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.

```

7838 \def\l@dcollect@@body#1\end#2{%
7839   \edef\l@dbegin@stack{\l@dpush@begins#1\begin\end
7840     \expandafter\@gobble\l@dbegin@stack}%
7841   \ifx\@empty\l@dbegin@stack
7842     \endgroup
7843     \@checkend{#2}%
7844     \addtol@denbody{#1}%
7845   \else
7846     \addtol@denbody{#1\end{#2}}%
7847   \fi
7848   \processl@denbody % A little tricky! Note the grouping
7849 }
7850
7851 %

```

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}{}`

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#2{%
  \@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>

```

## XXX.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`.

### XXX.2.1 Disabling and restoring commands

`\l@dtabnoexpands` More no expansion for critical and familiar footnotes in tabular environment.

```

7852 \newcommand*{\l@dtabnoexpands}{%
7853   \let\rtab=0%
7854   \let\ctab=0%
7855   \let\ltab=0%
7856   \let\rtabtext=0%
7857   \let\ltabtext=0%
7858   \let\ctabtext=0%
7859   \let\edbeforetab=0%
7860   \let\edaftertab=0%
7861   \let\edatleft=0%
7862   \let\edatright=0%
7863   \let\edvertline=0%
7864   \let\edvertdots=0%
7865   \let\edrowfill=0%
7866 }
7867
7868 %

```

`\disable@familiarnotes` Macros to disable and restore familiar notes, to prevent them from printing multiple times in `edtabularx` and `edarrayx` environments.

```

7869 \newcommand{\disable@familiarnotes}{%
7870   \unless\ifnofamiliar@%
7871     \def\do##1{%
7872       \csletcs{footnote@##1}{footnote##1}%

```



```

7873 \expandafter\renewcommand \csname footnote##1\endcsname[1]{%
7874 \protected@csxdef{@thefnmark##1}{\csuse{thefootnote##1}}%
7875 \csuse{@footnotemark##1}%
7876 }%
7877 }%
7878 \dolistloop{\@series}%
7879 \fi%
7880 }%
7881 \newcommand{\restore@familiarnotes}{%
7882 \unless\ifnofamiliar%
7883 \def\do##1{%
7884 \csletcs{footnote##1}{footnote@@##1}%
7885 }%
7886 \dolistloop{\@series}%
7887 \fi%
7888 }%
7889 %
7890 %

```

`\disable@sidenotes` The same, for side notes.

```

\restore@sidenotes
7891 \newcommand{\disable@sidenotes}{%
7892 \let\@@ledrightnote\ledrightnote%
7893 \let\@@ledleftnote\ledleftnote%
7894 \let\@@ledsidenote\ledsidenote%
7895 \let\ledrightnote@gobble%
7896 \let\ledleftnote@gobble%
7897 \let\ledsidenote@gobble%
7898 }%
7899 \newcommand{\restore@sidenotes}{%
7900 \let\ledrightnote\@@ledrightnote%
7901 \let\ledleftnote\@@ledleftnote%
7902 \let\ledsidenote\@@ledsidenote%
7903 }%
7904 %

```

`\disable@notes` Disable/restore side and familiar notes.

```

\restore@notes
7905 \newcommand{\disable@notes}{%
7906 \disable@sidenotes%
7907 \disable@familiarnotes%
7908 }%
7909 \newcommand{\restore@notes}{%
7910 \restore@sidenotes%
7911 \restore@familiarnotes%
7912 }%
7913 %

```

`\EDTEXT` We need to be able to modify the `\edtext` macros and also restore their original definitions.  
`\xedtext`

```

7914 \let\EDTEXT=\edtext
7915 \newcommand{\xedtext}[2]{\EDTEXT{#1}{#2}}
7916 %

```

**\EDLABEL** We need to be able to modify and restore the `\edlabel` macro.

```

\edlabel
7917 \let\EDLABEL=\edlabel
7918 \newcommand*{\xedlabel}[1]{\EDLABEL{#1}}
7919 %

```

**\xedindex** Macros supporting modification and restoration of `\edindex`.

```

\nulledindex
7920 \AtBeginDocument{\let\xedindex\edindex}%
7921 \newcommand{\nulledindex}[2][\jobname]{\@bsphack\@esphack}
7922
7923 %

```

**\@line@num** Macro supporting restoration of `\linenum`.

```

7924 \let\@line@num=\linenum
7925 %

```

**\l@dgobblearg** `\l@dgobbleoptarg[⟨arg⟩]{⟨arg⟩}` replaces these two arguments (first is optional) by `\relax`.

```

7926 \newcommand*{\l@dgobbleoptarg}[2][\relax]%
7927
7928 %

```

**\Relax** `\let\Relax=\relax`

**\NEXT** `\let\NEXT=\next`

```

7931
7932 %

```

**\l@dmodforedtext** **\l@drestoreforedtext** Modify and restore various macros for when `\edtext` is used.

```

7933 \newcommand{\l@dmodforedtext}{%
7934   \let\edtext\relax
7935   \def\do##1{\global\csletcs{##1footnote}{\l@dgobbleoptarg}}%
7936   \dolistloop{\@series}%
7937   \let\edindex\nulledindex
7938   \let\linenum@gobble}
7939 \newcommand{\l@drestoreforedtext}{%
7940   \def\do##1{\global\csletcs{##1footnote}{##1@footnote}}
7941   \dolistloop{\@series}%
7942   \let\edindex\xedindex}
7943 %

```

`\l@dnnullfills` Nullify and restore some column fillers, etc.

```
\l@drestorefills
7944 \newcommand{\l@dnnullfills}{%
7945   \def\edlabel##1{%
7946     \def\edrowfill##1##2##3{%
7947   }
7948 \newcommand{\l@drestorefills}{%
7949   \def\edrowfill##1##2##3{\@EDROWFILL@{##1}{##2}{##3}}%
7950 }
7951
7952 %
```

`\letsforverteilen` Gathers some lets and other code that is common to the *\*verteilen\** macros.

```
7953 \newcommand{\letsforverteilen}{%
7954   \let\edtext\xedtext
7955   \let\edindex\xedindex
7956   \def\do##1{\global\csletcs{##1footnote}{##1@@footnote}}
7957   \dolistloop{\@series}%
7958   \let\linenum@line@num
7959   \hilfsskip=\l@dcwidth%
7960   \advance\hilfsskip by -\wd\hilfsbox
7961   \def\edlabel##1{\xedlabel{##1}}
7962
7963 %
```

`\disablel@dtabfeet` Declarations for using or using `\edtext` inside tabulars. The default at this point is for  
`\enablel@dtabfeet` `\edtext`.

```
7964 \newcommand\disablel@dtabfeet{\l@dmodforedtext}%
7965 \newcommand\enablel@dtabfeet{\l@drestoreforedtext}%
7966 %
```

### XXX.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.

```
7967 \newcount\l@dampcount
7968 \l@dampcount=1\relax
7969 \newcount\l@dcolcount
7970 \l@dcolcount=0\relax
7971
7972 %
```

`\hilfsbox` Some (temporary) helper items.

```
\hilfsskip
7973 \newbox\hilfsbox
\Hilfsbox
7974 \newskip\hilfsskip
\hilfscount
7975 \newbox\Hilfsbox
```

```

7976 \newcount\hilfscount
7977
7978 %

```

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).

```

7979 \newdimen\dcoli
7980 \newdimen\dcolii
7981 \newdimen\dcoliii
7982 \newdimen\dcoliv
7983 \newdimen\dcolv
7984 \newdimen\dcolvi
7985 \newdimen\dcolvii
7986 \newdimen\dcolviii
7987 \newdimen\dcolix
7988 \newdimen\dcolx
7989 \newdimen\dcolxi
7990 \newdimen\dcolxii
7991 \newdimen\dcolxiii
7992 \newdimen\dcolxiv
7993 \newdimen\dcolxv
7994 \newdimen\dcolxvi
7995 \newdimen\dcolxvii
7996 \newdimen\dcolxviii
7997 \newdimen\dcolxix
7998 \newdimen\dcolxx
7999 \newdimen\dcolxxi
8000 \newdimen\dcolxxii
8001 \newdimen\dcolxxiii
8002 \newdimen\dcolxxiv
8003 \newdimen\dcolxxv
8004 \newdimen\dcolxxvi
8005 \newdimen\dcolxxvii
8006 \newdimen\dcolxxviii
8007 \newdimen\dcolxxix
8008 \newdimen\dcolxxx
8009 \newdimen\dcolerr % added for error handling
8010
8011 %

```

**\l@dcolwidth** This is a cunning way of storing the columnwidths indexed by the column number \l@dcolcount, like an array. (was \Dimenzuordnung)

```

8012 \newcommand{\l@dcolwidth}{\ifcase \the\l@dcolcount \dcoli %???
8013 \or \dcoli \or \dcolii \or \dcoliii
8014 \or \dcoliv \or \dcolv \or \dcolvi
8015 \or \dcolvii \or \dcolviii \or \dcolix \or \dcolx
8016 \or \dcolxi \or \dcolxii \or \dcolxiii
8017 \or \dcolxiv \or \dcolxv \or \dcolxvi

```

```

8018 \or \dcolxvii \or \dcolxviii \or \dcolxix \or \dcolxx
8019 \or \dcolxxi \or \dcolxxii \or \dcolxxiii
8020 \or \dcolxxiv \or \dcolxxv \or \dcolxxvi
8021 \or \dcolxxvii \or \dcolxxviii \or \dcolxxix \or \dcolxxx
8022 \else \dcolerr \fi}
8023
8024 %

```

**\stepl@dcolcount** This increments the column counter, and issues an error message if it is too large.

```

8025 \newcommand*\stepl@dcolcount{\advance\l@dcolcount\@ne
8026 \ifnum\l@dcolcount>30\relax
8027 \led@err@TooManyColumns
8028 \fi}
8029
8030 %

```

**\l@dsetmaxcolwidth** Sets the column width to the maximum value seen so far.

```

8031 \newcommand{\l@dsetmaxcolwidth}{%
8032 \ifdim\l@dcolwidth < \wd\hilfsbox
8033 \l@dcolwidth = \wd\hilfsbox
8034 \else \relax \fi}
8035
8036 %

```

**\measuremcell** Measure (recursively) the width required for a math cell.

```

8037 \def\measuremcell #1{%
8038 \ifx #1\ \ifnum\l@dcolcount=0\let\NEXT\relax%
8039 \else\l@dcheckcols%
8040 \l@dcolcount=0%
8041 \let\NEXT\measuremcell%
8042 \fi%
8043 \else\setbox\hilfsbox=\hbox{$\displaystyle{#1}$}%
8044 \stepl@dcolcount%
8045 \l@dsetmaxcolwidth%
8046 \let\NEXT\measuremcell%
8047 \fi\NEXT}
8048
8049 %

```

**\measuretcell** Measure (recursively) the width required for a text cell.

```

8050 \def\measuretcell #1{%
8051 \ifx #1\ \ifnum\l@dcolcount=0\let\NEXT\relax%
8052 \else\l@dcheckcols%
8053 \l@dcolcount=0%
8054 \let\NEXT\measuretcell%
8055 \fi%

```

```

8056 \else\setbox\hilfsbox=\hbox{#1}%
8057 \step1@dcolcount%
8058 \l@setmaxcolwidth%
8059 \let\NEXT\measuretcell%
8060 \fi\NEXT}
8061
8062 %

```

**\measuremrow** Measure (recursively) the width required for a math row.

```

8063 \def\measuremrow #1\{\%
8064 \ifx #1&\let\NEXT\relax%
8065 \else\measuremcell #1&\&\&\&%
8066 \let\NEXT\measuremrow%
8067 \fi\NEXT}
8068 %

```

**\measuretrow** Measure (recursively) the width required for a text row.

```

8069 \def\measuretrow #1\{\%
8070 \ifx #1&\let\NEXT\relax%
8071 \else\measuretcell #1&\&\&\&%
8072 \let\NEXT\measuretrow%
8073 \fi\NEXT}
8074
8075 %

```

**\edtabcolsep** The length `\edtabcolsep` controls the distance between columns.

```

8076 \newskip\edtabcolsep
8077 \global\edtabcolsep=10pt
8078
8079 %

```

```

\variab80 \newcommand{\variab}{\relax}
8081
8082 %

```

**\l@dcheckcols** Check that the number of columns is consistent.

```

8083 \newcommand*{\l@dcheckcols}{\%
8084 \ifnum\l@dcolcount=1\relax
8085 \else
8086 \ifnum\l@dampcount=1\relax
8087 \else
8088 \ifnum\l@dcolcount=\l@dampcount\relax
8089 \else
8090 \l@d@err@UnequalColumns

```

```

8091     \fi
8092     \fi
8093     \l@dampcount=\l@dcolcount
8094 \fi}
8095
8096 %

```

`\edfilldimen` A length.

```

8097 \newdimen\edfilldimen
8098 \edfilldimen=0pt
8099
8100 %

```

`\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`. We do not use the `\roman`  $\TeX$  command, because some packages, like `babel` can override it in some specific cases (Greek, for example).

`\theadcolcount`

```

8101 \newcounter{addcolcount}
8102 \renewcommand{\theadcolcount}{\romannumeral \c@addcolcount}
8103 %

```

### XXX.2.3 Tabular typesetting

`\setmcellright` Typeset (recursively) cells of display math right justified.

```

8104 \def\setmcellright #1&{\def\edlabel##1{}%
8105     \let\edindex\nulledindex
8106     \ifx #1\ \ifnum\l@dcolcount=0%\removelastskip
8107         \let\Next\relax%
8108     \else\l@dcolcount=0%
8109         \let\Next=\setmcellright%
8110     \fi%
8111 \else%
8112     \disablel@dtabfeet%
8113     \stepl@dcolcount%
8114     \disable@notes%
8115     \setbox\hilfsbox=\hbox{$\displaystyle{#1}$}%
8116     \restore@notes%
8117     \letsforverteilen%
8118     \hskip\hilfsskip$\displaystyle{#1}$%
8119     \hskip\edtabcolsep%
8120     \let\Next=\setmcellright%
8121 \fi\Next}
8122
8123 %

```

`\settcellright` Typeset (recursively) cells of text right justified.

```

8124 \def\settcellright #1&{\def\edlabel##1{}}%
8125 \let\edindex\nulledindex
8126 \ifx #1\\ \ifnum\l@dc@colcount=0%\removelastskip
8127 \let\Next\relax%
8128 \else\l@dc@colcount=0%
8129 \let\Next=\settcellright%
8130 \fi%
8131 \else%
8132 \disablel@dtabfeet%
8133 \step1@dc@colcount%
8134 \disable@notes%
8135 \setbox\hilfsbox=\hbox{#1}%
8136 \restore@notes%
8137 \letsforverteilen%
8138 \hskip\hilfsskip#1%
8139 \hskip\edtabcolsep%
8140 \let\Next=\settcellright%
8141 \fi\Next}
8142 %

```

**\setmcellleft** Typeset (recursively) cells of display math left justified.

```

8143 \def\setmcellleft #1&{\def\edlabel##1{}}%
8144 \let\edindex\nulledindex
8145 \ifx #1\\ \ifnum\l@dc@colcount=0 \let\Next\relax%
8146 \else\l@dc@colcount=0%
8147 \let\Next=\setmcellleft%
8148 \fi%
8149 \else \disablel@dtabfeet%
8150 \step1@dc@colcount%
8151 \disable@notes%
8152 \setbox\hilfsbox=\hbox{$\displaystyle{#1}$}%
8153 \restore@notes%
8154 \letsforverteilen%
8155 $\displaystyle{#1}$\hskip\hilfsskip\hskip\edtabcolsep%
8156 \let\Next=\setmcellleft%
8157 \fi\Next}
8158 %
8159 %

```

**\settcellleft** Typeset (recursively) cells of text left justified.

```

8160 \def\settcellleft #1&{\def\edlabel##1{}}%
8161 \let\edindex\nulledindex
8162 \ifx #1\\ \ifnum\l@dc@colcount=0 \let\Next\relax%
8163 \else\l@dc@colcount=0%
8164 \let\Next=\settcellleft%
8165 \fi%
8166 \else \disablel@dtabfeet%
8167 \step1@dc@colcount%

```



```

8168         \disable@notes%
8169         \setbox\hilfsbox=\hbox{#1}%
8170         \restore@notes%
8171         \letsforverteilen%
8172         #1\hskip\hilfsskip\hskip\edtabcolsep%
8173         \let\Next=\settcclleft%
8174     \fi\Next}
8175 %

```

**\setmcellcenter** Typeset (recursively) cells of display math centered.

```

8176 \def\setmcellcenter #1&{\def\edlabel##1}%
8177     \let\edindex\nulledindex
8178     \ifx #1\\ \ifnum\l@dc@count=0 \let\Next\relax%
8179         \else\l@dc@count=0%
8180             \let\Next=\setmcellcenter%
8181         \fi%
8182     \else \disablel@dtabfeet%
8183         \stepl@dc@count%
8184         \disable@notes%
8185         \setbox\hilfsbox=\hbox{$\displaystyle{#1}$}%
8186         \restore@notes%
8187         \letsforverteilen%
8188         \hskip 0.5\hilfsskip$\displaystyle{#1}$\hskip0.5\hilfsskip%
8189         \hskip\edtabcolsep%
8190         \let\Next=\setmcellcenter%
8191     \fi\Next}
8192 %
8193 %

```

**\settcclcenter** Typeset (recursively) cells of text centered.

```

8194 \def\settcclcenter #1&{\def\edlabel##1}%
8195     \let\edindex\nulledindex
8196     \ifx #1\\ \ifnum\l@dc@count=0 \let\Next\relax%
8197         \else\l@dc@count=0%
8198             \let\Next=\settcclcenter%
8199         \fi%
8200     \else \disablel@dtabfeet%
8201         \stepl@dc@count%
8202         \disable@notes%
8203         \setbox\hilfsbox=\hbox{#1}%
8204         \restore@notes%
8205         \letsforverteilen%
8206         \hskip 0.5\hilfsskip #1\hskip 0.5\hilfsskip%
8207         \hskip\edtabcolsep%
8208         \let\Next=\settcclcenter%
8209     \fi\Next}
8210 %
8211 %

```

```
\NEXT12 \let\NEXT=\relax
```

```
8213
```

```
8214 %
```

**\setmrowright** Typeset (recursively) rows of right justified math.

```
8215 \def\setmrowright #1\{\{%
8216   \ifx #1& \let\NEXT\relax
8217   \else \centerline{\setmcellright #1&\\&\\&}
8218     \let\NEXT=\setmrowright
8219   \fi\NEXT}
8220 %
```

**\settroright** Typeset (recursively) rows of right justified text.

```
8221 \def\settroright #1\{\{%
8222   \ifx #1& \let\NEXT\relax
8223   \else \centerline{\settcclright #1&\\&\\&}
8224     \let\NEXT=\settroright
8225   \fi\NEXT}
8226 %
8227 %
```

**\setmrowleft** Typeset (recursively) rows of left justified math.

```
8228 \def\setmrowleft #1\{\{%
8229   \ifx #1& \let\NEXT\relax
8230   \else \centerline{\setmcellleft #1&\\&\\&}
8231     \let\NEXT=\setmrowleft
8232   \fi\NEXT}
8233 %
```

**\settrorleft** Typeset (recursively) rows of left justified text.

```
8234 \def\settrorleft #1\{\{%
8235   \ifx #1& \let\NEXT\relax
8236   \else \centerline{\settcclleft #1&\\&\\&}
8237     \let\NEXT=\settrorleft
8238   \fi\NEXT}
8239 %
8240 %
```

**\setmrowcenter** Typeset (recursively) rows of centered math.

```
8241 \def\setmrowcenter #1\{\{%
8242   \ifx #1& \let\NEXT\relax%
8243   \else \centerline{\setmcellcenter #1&\\&\\&}
8244     \let\NEXT=\setmrowcenter
8245   \fi\NEXT}
8246 %
```

**\settrowcenter** Typeset (recursively) rows of centered text.

```

8247 \def\settrowcenter #1\{\%
8248   \ifx #1& \let\NEXT\relax
8249   \else \centerline{\settcellcenter #1&\&\&}
8250   \let\NEXT=\settrowcenter
8251   \fi\NEXT}
8252
8253 %

```

```

\newcommand{\nullsetzen}%
8255   \step1@dcolcount%
8256   \l@dcolwidth=0pt%
8257   \ifnum\l@dcolcount=30\let\NEXT\relax%
8258   \l@dcolcount=0\relax
8259   \else\let\NEXT\nullsetzen%
8260   \fi\NEXT}
8261
8262 %

```

**\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.

```

8263 \newcommand{\edatleft}[3][\@empty]{%
8264   \ifx#1\@empty
8265     \vbox to 10pt{\vss\hbox{\$ \left#2\vrule width0pt height #3
8266       depth 0pt \right. \$\hss}\vfil}
8267   \else
8268     \vbox to 4pt{\vss\hbox{\$#1\left#2\vrule width0pt height #3
8269       depth 0pt \right. \$}\vfil}
8270   \fi}
8271 %

```

**\edatright**  $\edatright[\langle math \rangle]{\langle symbol \rangle}{\langle len \rangle}$ . Right  $\langle symbol \rangle$ ,  $2\langle len \rangle$  high with appended  $\langle math \rangle$  vertically centered.

```

8272 \newcommand{\edatright}[3][\@empty]{%
8273   \ifx#1\@empty
8274     \vbox to 10pt{\vss\hbox{\$ \left.\vrule width0pt height #3
8275       depth 0pt \right#2 \$\hss}\vfil}
8276   \else
8277     \vbox to 4pt{\vss\hbox{\$ \left.\vrule width0pt height #3
8278       depth 0pt \right#2 #1 \$}\vfil}
8279   \fi}
8280
8281 %

```

**\edvertline**  $\edvertline{\langle len \rangle}$  vertical line  $\langle len \rangle$  high.

```

8282 \newcommand{\edvertline}[1]{\vbox to 8pt{\vss\hbox{\vrule height #1}\vfil}}
8283
8284 %

```

**\edvertdots** `\edvertdots{<len>}` vertical dotted line <len> high.

```

8285 \newcommand{\edvertdots}[1]{\vbox to 1pt{\vss\vbox to #1%
8286   {\cleaders\hbox{$\math\hbox{.}\vbox to 0.5em{ }$}\vfil}}}
8287
8288 %

```

**\l@dtabaddcols** `\l@dtabaddcols{<startcol>}{<endcol>}` adds the widths of the columns <startcol> through <endcol> to `\edfilldimen`. It is a  $\TeX$  style reimplementation of the original `\@add@`.

```

8289 \newcommand{\l@dtabaddcols}[2]{%
8290   \l@dcheckstartend{#1}{#2}%
8291   \ifl@dstartendok
8292     \setcounter{addcolcount}{#1}%
8293     \@whilenum \value{addcolcount}<#2\relax \do
8294       {\advance\edfilldimen by \the \csname dcol\theaddcolcount\endcsname
8295        \advance\edfilldimen by \edtabcolsep
8296        \stepcounter{addcolcount}}%
8297     \advance\edfilldimen by \the \csname dcol\theaddcolcount\endcsname
8298     \fi
8299   }
8300
8301 %

```

**\ifl@dstartendok** `\l@dcheckstartend{<startcol>}{<endcol>}` checks that the values of <startcol> and <endcol> are sensible. If they are then `\ifl@dstartendok` is set TRUE, otherwise it is set FALSE.

```

8302 \newif\ifl@dstartendok
8303 \newcommand{\l@dcheckstartend}[2]{%
8304   \l@dstartendoktrue
8305   \ifnum #1<\@ne
8306     \l@dstartendokfalse
8307     \led@err@LowStartColumn
8308   \fi
8309   \ifnum #2>30\relax
8310     \l@dstartendokfalse
8311     \led@err@HighEndColumn
8312   \fi
8313   \ifnum #1>#2\relax
8314     \l@dstartendokfalse
8315     \led@err@ReverseColumns
8316   \fi
8317 }
8318
8319 %

```

`\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.

```

8320 \newcommand*{\edrowfill}[3]{%
8321   \l@dtabaddcols{#1}{#2}%
8322   \hb@xt@ \the\l@dcolwidth{\hb@xt@ \the\edfilldimen{#3}\hss}}
8323 \let\@edrowfill=\edrowfill
8324 \def\@EDROWFILL@#1#2#3{\@edrowfill@{#1}{#2}{#3}}
8325
8326 %

```

`\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`.

```

8327 \newcommand{\leftltab}[1]{%
8328   \hb@xt@ \z@{\vbox{\edtabindent%
8329     \moveleft\Hilfsskip\hbox{\ #1}}\hss}}
8330
8331 %

```

`\leftrtab` `\leftrtab{<text>}{<math>}` for `\edbeforetab` in `\rtab`.

```

8332 \newcommand{\leftrtab}[2]{%
8333   #2\hb@xt@ \z@{\vbox{\edtabindent%
8334     \advance\Hilfsskip by \dcoli%
8335     \moveleft\Hilfsskip\hbox{\ #1}}\hss}}
8336
8337 %

```

`\leftctab` `\leftctab{<text>}{<math>}` for `\edbeforetab` in `\ctab`.

```

8338 \newcommand{\leftctab}[2]{%
8339   \hb@xt@ \z@{\vbox{\edtabindent\l@dcolcount=\l@dampcount%
8340     \advance\Hilfsskip by 0.5\dcoli%
8341     \setbox\hilfsbox=\hbox{\def\edlabel##1{%
8342       \disablel@dtabfeet$\displaystyle{#2}$}%
8343     \advance\Hilfsskip by -0.5\wd\hilfsbox%
8344     \moveleft\Hilfsskip\hbox{\ #1}}\hss}%
8345     #2}
8346
8347 %

```

`\rightctab` `\rightctab{<math>}{<text>}` for `\edaftertab` in `\ctab`.

```

8348 \newcommand{\rightctab}[2]{%
8349     \setbox\hilfsbox=\hbox{\def\edlabel##1{%
8350         \disablel@dtabfeet#2}\l@dampcount=\l@dcolcount%
8351         #1\hb@xt@\z@\vbox{\edtabindent\l@dcolcount=\l@dampcount%
8352         \advance\Hilfsskip by 0.5\l@dcolwidth%
8353         \advance\Hilfsskip by -\wd\hilfsbox%
8354         \setbox\hilfsbox=\hbox{\def\edlabel##1{%
8355             \disablel@dtabfeet$\displaystyle{#1}$}%
8356             \advance\Hilfsskip by -0.5\wd\hilfsbox%
8357             \advance\Hilfsskip by \edtabcolsep%
8358             \moveright\Hilfsskip\hbox{ #2}}\hss}%
8359         }
8360     }
8361 %

```

**\rightltab** `\rightltab{<math><math>}{<text>}` for `\edaftertab` in `\ltab`.

```

8362 \newcommand{\rightltab}[2]{%
8363     \setbox\hilfsbox=\hbox{\def\edlabel##1{%
8364         \disablel@dtabfeet#2}\l@dampcount=\l@dcolcount%
8365         #1\hb@xt@\z@\vbox{\edtabindent\l@dcolcount=\l@dampcount%
8366         \advance\Hilfsskip by \l@dcolwidth%
8367         \advance\Hilfsskip by -\wd\hilfsbox%
8368         \setbox\hilfsbox=\hbox{\def\edlabel##1{%
8369             \disablel@dtabfeet$\displaystyle{#1}$}%
8370             \advance\Hilfsskip by -\wd\hilfsbox%
8371             \advance\Hilfsskip by \edtabcolsep%
8372             \moveright\Hilfsskip\hbox{ #2}}\hss}%
8373         }
8374     }
8375 %

```

**\rightrtab** `\rightrtab{<math><math>}{<text>}` for `\edaftertab` in `\rtab`.

```

8376 \newcommand{\rightrtab}[2]{%
8377     \setbox\hilfsbox=\hbox{\def\edlabel##1{%
8378         \disablel@dtabfeet#2}%
8379         #1\hb@xt@\z@\vbox{\edtabindent%
8380         \advance\Hilfsskip by -\wd\hilfsbox%
8381         \advance\Hilfsskip by \edtabcolsep%
8382         \moveright\Hilfsskip\hbox{ #2}}\hss}%
8383     }
8384 %
8385 %

```

**\rtab** `\rtab{<body>}` typesets `<body>` as an array with the entries right justified.

**\edbeforetab** The process is first to measure the `<body>` to get the column widths, and then in a  
**\edaftertab** second pass to typeset the body.

```

8386 \newcommand{\rtab}[1]{%
8387   \l@dnnullfills
8388   \def\edbeforetab##1##2{\lefttrtab{##1}{##2}}%
8389   \def\edaftertab##1##2{\righttrtab{##1}{##2}}%
8390   \measurebody{#1}%
8391   \l@drestorefills
8392   \variab
8393   \setmrowright #1\&\&\&%
8394   \enablel@dtabfeet}
8395
8396 %

```

**\measurebody** `\measurebody{<body>}` measures the array `<body>`.

```

8397 \newcommand{\measurebody}[1]{%
8398   \disablel@dtabfeet%
8399   \l@dcolcount=0%
8400   \nullsetzen%
8401   \l@dcolcount=0
8402   \measuremrow #1\&\&\&%
8403   \global\l@dampcount=1}
8404
8405 %

```

**\rtabtext** `\rtabtext{<body>}` typesets `<body>` as a tabular with the entries right justified.

```

8406 \newcommand{\rtabtext}[1]{%
8407   \l@dnnullfills
8408   \measuretbody{#1}%
8409   \l@drestorefills
8410   \variab
8411   \settroright #1\&\&\&%
8412   \enablel@dtabfeet}
8413
8414 %

```

**\measuretbody** `\measuretbody{<body>}` measures the tabular `<body>`.

```

8415 \newcommand{\measuretbody}[1]{%
8416   \disable@notes%
8417   \disablel@dtabfeet%
8418   \l@dcolcount=0%
8419   \nullsetzen%
8420   \l@dcolcount=0
8421   \measuretror #1\&\&\&%
8422   \restore@notes%
8423   \global\l@dampcount=1}
8424
8425 %

```

`\ltab` Array with entries left justified.

```
\edbeforetab
\edaftertab
8426 \newcommand{\ltab}[1]{%
8427   \l@dnnullfills
8428   \def\edbeforetab##1##2{\leftltab{##1}{##2}}%
8429   \def\edaftertab##1##2{\rightltab{##1}{##2}}%
8430   \measurebody{#1}%
8431   \l@drestorefills
8432   \variab
8433   \setmrowleft #1\\&\\%
8434   \enablel@dtabfeet}
8435
8436 %
```

`\ltabtext` Tabular with entries left justified.

```
8437 \newcommand{\ltabtext}[1]{%
8438   \l@dnnullfills
8439   \measuretbody{#1}%
8440   \l@drestorefills
8441   \variab
8442   \settrrowleft #1\\&\\%
8443   \enablel@dtabfeet}
8444
8445 %
```

`\ctab` Array with centered entries.

```
\edbeforetab
\edaftertab
8446 \newcommand{\ctab}[1]{%
8447   \l@dnnullfills
8448   \def\edbeforetab##1##2{\leftctab{##1}{##2}}%
8449   \def\edaftertab##1##2{\rightctab{##1}{##2}}%
8450   \measurebody{#1}%
8451   \l@drestorefills
8452   \variab
8453   \setmrowcenter #1\\&\\%
8454   \enablel@dtabfeet}
8455
8456 %
```

`\ctabtext` Tabular with entries centered.

```
8457 \newcommand{\ctabtext}[1]{%
8458   \l@dnnullfills
8459   \measuretbody{#1}%
8460   \l@drestorefills
8461   \variab
8462   \settrrowcenter #1\\&\\%
8463   \enablel@dtabfeet}
8464
8465 %
```



```
\spreadtext8466 \newcommand{\spreadtext}[1]{%\l@dcolcount=\l@dampcount%
8467 \hb@xt@ \the\l@dcolwidth{\hbox{#1}\hss}}
8468 %
```

```
\spreadmath8469 \newcommand{\spreadmath}[1]{%
8470 \hb@xt@ \the\l@dcolwidth{\hbox{$\displaystyle{#1}$}\hss}}
8471
8472 %
```

\HILFSskip More helpers.

```
\Hilfsskip8473 \newskip\HILFSskip
8474 \newskip\Hilfsskip
8475
8476 %
```

```
\EDTABINDENT8477 \newcommand{\EDTABINDENT}{%
8478 \ifnum\l@dcolcount=30\let\NEXT\relax\l@dcolcount=0%
8479 \else\step\l@dcolcount%
8480 \advance\Hilfsskip by\l@dcolwidth%
8481 \ifdim\l@dcolwidth=0pt\advance\hilfscount\@ne
8482 \else\advance\Hilfsskip by \the\hilfscount\edtabcolsep%
8483 \hilfscount=1\fi%
8484 \let\NEXT=\EDTABINDENT%
8485 \fi\NEXT}%
8486 %
```

\edtabindent (was \tabindent)

```
8487 \newcommand{\edtabindent}{%
8488 \l@dcolcount=0\relax
8489 \Hilfsskip=0pt%
8490 \hilfscount=1\relax
8491 \EDTABINDENT%
8492 \hilfsskip=\hsize%
8493 \advance\hilfsskip -\Hilfsskip%
8494 \Hilfsskip=0.5\hilfsskip%
8495 }%
8496
8497 %
```

\EDTAB (was \TAB)

```
8498 \def\EDTAB #1|#2|{%
8499 \setbox\tabhilfbox=\hbox{$\displaystyle{#1}$}%
8500 \setbox\tabHilfbox=\hbox{$\displaystyle{#2}$}%
8501 \advance\tabelskip -\wd\tabhilfbox%
```

`\EDTABtext` (was `\TABtext`)

`\tabhilfbox` Further helpers.

edarrayl edarrayc edarrayr The ‘environment’ forms for \ltab, \ctab and \rtab.

edtabularl edtabularc edtabularr The ‘environment’ forms for `\ltabtext`, `\ctabtext` and `\rtabtext`.

## XXXI Quotation's commands

`\initnumbering@quote` This macro, called at the beginning of any numbered section, locally redefines the quotation and quote environments, in order to allow their use inside of numbered sections.

```

\quotation \initnumbering@quote defines quotation environment.
\endquotation
\quote
\endquote
\newcommand{\initnumbering@quote}{
  \ifnoquotation@else
  \renewcommand{\quotation}{\par\leavevmode%
    \parindent=1.5em%
    \skipnumbering%
    \ifautopar%
      \vskip-\parskip%
    \else%
      \vskip\topsep%
    \fi%
    \global\leftskip=\leftmargin%
    \global\rightskip=\leftmargin%
  }
  \renewcommand{\endquotation}{\par%
    \global\leftskip=0pt%
    \global\rightskip=0pt%
    \leavevmode%
    \skipnumbering%
    \ifautopar%
      \vskip-\parskip%
    \else%
      \vskip\topsep%
    \fi%
  }
  \renewcommand{\quote}{\par\leavevmode%
    \parindent=0pt%
    \skipnumbering%
    \ifautopar%
      \vskip-\parskip%
    \else%
      \vskip\topsep%
    \fi%
    \global\leftskip=\leftmargin%
    \global\rightskip=\leftmargin%
  }
  \renewcommand{\endquote}{\par%
    \global\leftskip=0pt%
    \global\rightskip=0pt%
    \leavevmode%
    \skipnumbering%
    \ifautopar%
      \vskip-\parskip%
    \else%
      \vskip\topsep%
    \fi%
  }
}
\fi
}

```

```
8577 %
```

## XXXII Section's title commands

### XXXII.1 Commands to disable some feature

`\ledsectnotoc` The `\ledsectnotoc` only disables the `\addcontentsline` macro.

```
8578 \newcommand{\ledsectnotoc}{\let\addcontentsline\@gobblethree}
8579 %
```

`\ledsectnomark` The `\ledsectnomark` only disables the `\chaptermark`, `\sectionmark` and `\subsectionmark` macros.

```
8580 \newcommand{\ledsectnomark}{%
8581   \let\chaptermark\@gobble%
8582   \let\sectionmark\@gobble%
8583   \let\subsectionmark\@gobble%
8584 }
8585 %
```

### XXXII.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.

### XXXII.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).

```
8586 \catcode`\#=12
8587 \notbool{@noeled@sec}{%
8588 %
```

`\beforeeledchapter` For technical reasons, not yet solved, page-breaking before chapters can't be made automatically by `eledmac`. Users have to use `\beforeeledchapter`.

```
8589 \ifl@dmemoir
8590   \newcommand\beforeeledchapter{%
8591     \clearforchapter%
8592   }
8593 \else
8594   \newcommand\beforeeledchapter{%
8595     \if@openright%
8596       \cleardoublepage%
8597     \else%
8598       \clearpage%
8599     \fi%
8600   }
8601 \fi
8602 %
```

### XXXII.4 Auxiliary commands

`\print@leftmargin@eledsection` `\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.

```
8603 \def\print@rightmargin@eledsection{%
8604   \if@eled@sectioning%
8605     \begingroup%
8606     \if@RTL%
8607       \let\llap\rlap%
8608       \let\leftlinenum\rightlinenum%
8609       \let\leftlinenumR\rightlinenumR%
8610       \let\l@drd@ta\l@dld@ta%
8611       \let\l@drsn@te\l@dlsn@te%
8612     \fi%
8613     \hfill\l@drd@ta \csuse{LR}{\l@drsn@te}%
8614   \endgroup%
8615 \fi%
```

```

8616 }%
8617
8618 \def\print@leftmargin@eledsection{%
8619   \if@eled@sectioning%
8620     \leavevmode%
8621     \begingroup%
8622     \if@RTL%
8623       \let\rlap\llap%
8624       \let\rightlinenum\leftlinenum%
8625       \let\rightlinenumR\leftlinenumR%
8626       \let\l@dld@ta\l@drd@ta%
8627       \let\l@dlsn@te\l@drsn@te%
8628     \fi%
8629     \l@dld@ta\csuse{LR}{\l@dlsn@te}%
8630     \endgroup%
8631   \fi%
8632 }%
8633
8634 %

```

### XXXII.5 Patching standard commands

We have to patch  $\LaTeX$ , book and memoir sectioning commands in order to:

- Disable `\edtext` inside.
- Disable page breaking (for `\chapter`).
- Add line numbers and sidenotes.

Unfortunately, Maïeul Rouquette was not able to try if memoir is loaded. That is why `eledmac` tries to define for both standard class and memoir class.

```

8635 \AtBeginDocument{%
8636
8637
8638 \pretocmd{\M@sect}
8639   {\let\old@edtext=\edtext%
8640    \let\edtext=\dummy@edtext@showlemma%
8641   }
8642   {}
8643   {}
8644
8645 \apptocmd{\M@sect}
8646   {\let\edtext=\old@edtext}
8647   {}
8648   {}
8649
8650 \patchcmd{\M@sect}
8651   { #9}

```

```

8652 { #9%
8653 \print@rightmargin@eledsection%
8654 }
8655 {}
8656 {}
8657
8658 \patchcmd{\M@sect}
8659 {\hskip #3\relax}
8660 {\hskip #3\relax%
8661 \print@leftmargin@eledsection%
8662 }
8663 {}
8664 {}
8665
8666 \patchcmd{\@mem@old@ssect}
8667 {#5}
8668 {#5%
8669 \print@leftmargin@eledsection%
8670 }
8671 {}
8672 {}
8673
8674 \patchcmd{\@mem@old@ssect}
8675 {\hskip #1}
8676 {\hskip #1%
8677 \print@rightmargin@eledsection%
8678 }
8679 {}
8680 {}
8681
8682
8683
8684 \patchcmd{\scr@startchapter}{\if@openright\cleardoublepage\else\clearpage\fi}{%
8685 \if@eled@sectioning\else%
8686 \ifl@dprintingpages\else%
8687 \if@openright\cleardoublepage\else\clearpage\fi%No clearpage inside a
8688 \Pages: will keep critical notes from printing on the title page. Here for
8689 scrbook.
8690 \fi%
8691 \fi%
8692 }
8693 {}
8694 {}
8695
8696 \patchcmd{\@makechapterhead}
8697 {#1}
8698 {\print@leftmargin@eledsection%
8699 #1%
8700 \print@rightmargin@eledsection%

```

```

8699 }
8700 {}
8701 {}
8702
8703 \patchcmd{\@makechapterhead}% For BIDI
8704 {\if@RTL\raggedleft\else\raggedright\fi}%
8705 {\if@eled@sectioning\else%
8706   \if@RTL\raggedleft\else\raggedright\fi%
8707   \fi%
8708 }%
8709 {}%
8710 {}%
8711
8712 \patchcmd{\@makeschapterhead}
8713   {\#1}
8714   {\print@leftmargin@eledsection%
8715     \#1%
8716     \print@rightmargin@eledsection%
8717   }
8718   {}
8719   {}
8720
8721 \pretocmd{\@sect}
8722   {\let\old@edtext=\edtext
8723     \let\edtext=\dummy@edtext@showlemma%
8724   }
8725   {}
8726   {}
8727
8728 \apptocmd{\@sect}
8729   {\let\edtext=\old@edtext}
8730   {}
8731   {}
8732
8733 \pretocmd{\@ssect}
8734   {\let\old@edtext=\edtext%
8735     \let\edtext=\dummy@edtext@showlemma%
8736   }
8737   {}
8738   {}
8739
8740 \apptocmd{\@ssect}
8741   {\let\edtext=\old@edtext}
8742   {}
8743   {}
8744
8745 %

```

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.



```

8746 \@ifpackageloaded{nameref}{
8747
8748   \patchcmd{\NR@sect}
8749     {#8}
8750     {#8%
8751       \print@rightmargin@eledsection%
8752     }
8753     {}
8754     {}
8755
8756   \patchcmd{\NR@sect}
8757     {\hskip #3\relax}
8758     {\hskip #3\relax%
8759       \print@leftmargin@eledsection%
8760     }
8761     {}
8762     {}
8763
8764   \patchcmd{\NR@ssect}
8765     {#5}
8766     {#5%
8767       \print@rightmargin@eledsection%
8768     }
8769     {}
8770     {}
8771
8772   \patchcmd{\NR@ssect}
8773     {\hskip #1}
8774     {\hskip #1%
8775       \print@leftmargin@eledsection%
8776     }
8777     {}
8778     {}
8779   }%
8780   {
8781     \patchcmd{\@sect}
8782       {#8}
8783       {#8%
8784         \print@rightmargin@eledsection%
8785       }
8786       {}
8787       {}
8788
8789     \patchcmd{\@sect}
8790       {\hskip #3\relax}
8791       {\hskip #3\relax%
8792         \print@leftmargin@eledsection%
8793       }
8794       {}
8795       {}

```

```

8796 \patchcmd{\@ssect}
8797   {\#5}
8798   {\#5%
8799   \print@rightmargin@eledsection%
8800   }
8801   {}
8802   {}
8803   {}
8804
8805 \patchcmd{\@ssect}
8806   {\hskip #1}
8807   {\hskip #1%
8808   \print@leftmargin@eledsection%
8809   }
8810   {}
8811   {}
8812   }%
8813 }%
8814 %

```

Close the `\notbool{@noeled@sec}` statement. Also, we have finished patching the commands, using `#` with a catcode equal to 12, so we are restoring the normal catcode for `#`.

```

8815 {}}%
8816 \protect\catcode`\#=6 %Space NEEDS by \catcode
8817 %

```

**\chapter** We patch the `\chapter` command even if the `noeledsec` option is called, because we can use `\chapter` in the optional argument of a `\pstart` in parallel typesetting.

```

8818 \AtBeginDocument{%
8819 \patchcmd{\chapter}{\clearforchapter}{%
8820   \if@eled@sectioning\else%
8821   \ifl@dprintingpages\else%
8822   \clearforchapter%
8823   \fi%
8824   \fi%
8825   }%
8826   {}%
8827   {}%
8828
8829 \patchcmd{\chapter}{\if@openright\cleardoublepage\else\clearpage\fi}{%
8830   \if@eled@sectioning\else%
8831   \ifl@dprintingpages%
8832   \endgraf%
8833   \else%
8834   \if@openright\cleardoublepage\else\clearpage\fi%No clearpage inside a
\Pages: will keep critical notes from printing on the title page. Here for
classical classes
8835   \fi%

```

```

8836 \fi%
8837 }%
8838 {}%
8839 {}%
8840 }%
8841 %

```

**\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.

```

8842 \newif\if@eled@sectioning%
8843 %

```

We reopen a new \notbool{@noeled@sec} statement, as we will define the \elesection commands.

```

8844 \notbool{@noeled@sec}{%
8845 %

```

## XXXII.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.

```

8846 \newwrite\eled@sectioning@out
8847 %

```

**\eledchapter** And now, the user sectioning commands, which write to the file, and also add content as a “normal” line.

```

\eledsection
\eledsubsection
\eledsubsubsection
\eledchapter*
\eledsection*
\eledsubsection*
\eledsubsubsection*
8848 \newcommand{\eledchapter}[2] [] {%
8849 \disable@familiarnotes%
8850 #2%
8851 \restore@familiarnotes%
8852 \ifledRcol%
8853 \immediate\write\eled@sectioningR@out{%
8854 \string\eled@chapter{#1}{\unexpanded{#2}}{\the\l@dnumpstartsR}}{R}
8855 }%
8856 \else%
8857 \immediate\write\eled@sectioning@out{%
8858 \string\eled@chapter{#1}{\unexpanded{#2}}{\the\l@dnumpstartsL}}{L}
8859 }%
8860 \fi%
8861 }
8862
8863 \newcommand{\eledsection}[2] [] {%
8864 \disable@familiarnotes%
8865 #2%

```

```

8866 \restore@familiarnotes%
8867 \ifledRcol%
8868 \immediate\write\eled@sectioningR@out{%
8869 \string\eled@section{#1}{\unexpanded{#2}}{\the\l@dnumpstartsR}{R}
8870 }%
8871 \else%
8872 \immediate\write\eled@sectioning@out{%
8873 \string\eled@section{#1}{\unexpanded{#2}}{\the\l@dnumpstartsL}{L}
8874 }%
8875 \fi%
8876 }
8877
8878 \newcommand{\eledsubsection}[2][]{%
8879 \disable@familiarnotes%
8880 #2%
8881 \restore@familiarnotes%
8882 \ifledRcol%
8883 \immediate\write\eled@sectioningR@out{%
8884 \string\eled@subsection{#1}{\unexpanded{#2}}{\the\l@dnumpstartsR}{R}
8885 }%
8886 \else%
8887 \immediate\write\eled@sectioning@out{%
8888 \string\eled@subsection{#1}{\unexpanded{#2}}{\the\l@dnumpstartsL}{L}
8889 }%
8890 \fi%
8891 }
8892 \newcommand{\eledsubsubsection}[2][]{%
8893 \disable@familiarnotes%
8894 #2%
8895 \restore@familiarnotes%
8896 \ifledRcol%
8897 \immediate\write\eled@sectioningR@out{%
8898 \string\eled@subsubsection{#1}{\unexpanded{#2}}{\the\l@dnumpstartsR}
8899 }{R}
8900 }%
8901 \else%
8902 \immediate\write\eled@sectioning@out{%
8903 \string\eled@subsubsection{#1}{\unexpanded{#2}}{\the\l@dnumpstartsL}
8904 }{L}
8905 }%
8906 \fi%
8907 }
8908
8909 \WithSuffix\newcommand{\eledchapter}[2][]{%
8910 \disable@familiarnotes%
8911 #2%
8912 \restore@familiarnotes%
8913 \ifledRcol%

```

```

8913 \immediate\write\eled@sectioningR@out{%
8914 \string\eled@chapter{#1}{\unexpanded{#2}}{\the\l@dnumpstartsR}{*}{R}
8915 }%
8916 \else%
8917 \immediate\write\eled@sectioning@out{%
8918 \string\eled@chapter{#1}{\unexpanded{#2}}{\the\l@dnumpstartsL}{*}{L}
8919 }%
8920 \fi%
8921 }
8922
8923 \WithSuffix\newcommand\eledsection*[2] [] {%
8924 \disable@familiarnotes%
8925 #2%
8926 \restore@familiarnotes%
8927 \ifledRcol%
8928 \immediate\write\eled@sectioningR@out{%
8929 \string\eled@section{#1}{\unexpanded{#2}}{\the\l@dnumpstartsR}{*}{R}
8930 }%
8931 \else%
8932 \immediate\write\eled@sectioning@out{%
8933 \string\eled@section{#1}{\unexpanded{#2}}{\the\l@dnumpstartsL}{*}{L}
8934 }%
8935 \fi%
8936 }
8937
8938 \WithSuffix\newcommand\eledsubsection*[2] [] {%
8939 \disable@familiarnotes%
8940 #2%
8941 \restore@familiarnotes%
8942 \ifledRcol%
8943 \immediate\write\eled@sectioningR@out{%
8944 \string\eled@subsection{#1}{\unexpanded{#2}}{\the\l@dnumpstartsR}{*}{R}
8945 }%
8946 \else%
8947 \immediate\write\eled@sectioning@out{%
8948 \string\eled@subsection{#1}{\unexpanded{#2}}{\the\l@dnumpstartsL}
8949 }{*}{L}
8950 \fi%
8951 }
8952
8953 \WithSuffix\newcommand\eledsubsubsection*[2] [] {%
8954 \disable@familiarnotes%
8955 #2%
8956 \restore@familiarnotes%
8957 \ifledRcol%
8958 \immediate\write\eled@sectioningR@out{%
8959 \string\eled@subsubsection{#1}{\unexpanded{#2}}{\the\l@dnumpstartsR}
8960 }{*}{R}

```

```

8960 }%
8961 \else%
8962   \immediate\write\eled@sectioning@out{%
8963     \string\eled@subsubsection{#1}{\unexpanded{#2}}{\the\l@dumpstartsL
8964   }{*}{}}
8965   }%
8966 \fi%
8967 }
%
```

### XXXII.7 Macros written in the auxiliary file

`\eled@chapter`  
`\eled@section`  
`\eled@subsection`  
`\eled@subsubsection`

The sectioning macros, called in the auxiliary file. They have five arguments:

1. Optional arguments of  $\LaTeX$  sectioning command.
2. Mandatory arguments of  $\LaTeX$  sectioning command.
3. Pstart number.
4. Side: R if right, nothing if left.
5. Starred or not.

```

8968 \def\eled@chapter#1#2#3#4#5{%
8969   \ifstrempy{#4}%
8970   {%
8971     \ifstrempy{#1}%
8972     {%
8973       \csgdef{eled@sectioning@#3#5}{\let\edtext=\dummy@edtext@showlemma\
chapter{#2}}%
8974       \csgdef{eled@sectmark@#3#5}{\let\edtext=\dummy@edtext{}\chaptermark
{#2}}%
8975       }%Need for \pairs, because of using parbox.
8976       {%
8977         \csgdef{eled@sectioning@#3#5}{\let\edtext=\dummy@edtext@showlemma\
chapter[#1]{#2}}%
8978         \csgdef{eled@sectmark@#3#5}{\let\edtext=\dummy@edtext{}\chaptermark
{#2}}}%Need for \pairs, because of using parbox.
8979       }%
8980     }%
8981     {%
8982       \ifstrempy{#1}%
8983       {\csgdef{eled@sectioning@#3#5}{\let\edtext=\dummy@edtext@showlemma\
chapter*{#2}}}%
8984       {\csgdef{eled@sectioning@#3#5}{\let\edtext=\dummy@edtext@showlemma\
chapter*[#1]{#2}}}%Bug in LaTeX!
8985     }%
8986     \listcsgadd{eled@sections#5@@}{#3}%
8987   }
}
```

```

8988 \def\eled@section#1#2#3#4#5{%
8989     \ifstrempy{#4}%
8990     {\ifstrempy{#1}%
8991     {%
8992         \csgdef{eled@sectioning@#3#5}{\section{#2}}%
8993         \csgdef{eled@sectmark@#3#5}{\let\edtext=\dummy@edtext{}\sectionmark
{\#2}}}%Need for \pairs, because of using parbox.
8994     }%
8995     {%
8996         \csgdef{eled@sectioning@#3#5}{\section[#1]{#2}}%
8997         \csgdef{eled@sectmark@#3#5}{\let\edtext=\dummy@edtext{}\sectionmark
{\#1}}}%Need for \pairs, because of using parbox.
8998     }%
8999 }%
9000 {\ifstrempy{#1}%
9001     {\csgdef{eled@sectioning@#3#5}{\section*{#2}}}%
9002     {\csgdef{eled@sectioning@#3#5}{\section*{#1}{#2}}}%Bug in LaTeX!
9003 }
9004 \listcsgadd{eled@sections#5@@}{#3}%
9005 }
9006 \def\eled@subsection#1#2#3#4#5{%
9007     \ifstrempy{#4}%
9008     {\ifstrempy{#1}%
9009     {%
9010         \csgdef{eled@sectioning@#3#5}{\subsection{#2}}%
9011         \csgdef{eled@sectmark@#3#5}{\let\edtext=\dummy@edtext{}\csuse{
subsectionmark}{#2}}}%Need for \pairs, because of using parbox. \csuse in
case of \subsectionmark is not defined (book)
9012     }%
9013     {%
9014         \csgdef{eled@sectioning@#3#5}{\subsection[#1]{#2}}%
9015         \csgdef{eled@sectmark@#3#5}{\let\edtext=\dummy@edtext{}\csuse{
subsectionmark}{#1}}}%Need for \pairs, because of using parbox. \csuse in
case of \subsectionmark is not defined (book)
9016     }%
9017 }%
9018 {\ifstrempy{#1}%
9019     {\csgdef{eled@sectioning@#3#5}{\subsection*{#2}}}%
9020     {\csgdef{eled@sectioning@#3#5}{\subsection*{#1}{#2}}}%Bug in LaTeX!
9021 }
9022 \listcsgadd{eled@sections#5@@}{#3}%
9023 }
9024 \def\eled@subsubsection#1#2#3#4#5{%
9025     \ifstrempy{#4}%
9026     {\ifstrempy{#1}%
9027     {%
9028         \csgdef{eled@sectioning@#3#5}{\subsubsection{#2}}%
9029         \csgdef{eled@sectioning@#3#5}{\subsubsection[#1]{#2}}%
9029     }%
9030     {\ifstrempy{#1}%
9031         {\csgdef{eled@sectioning@#3#5}{\subsubsection*{#2}}}%

```

```

9032      {\csgdef{eled@sectioning@#3#5}{\subsubsection*[#1]{#2}}}%Bug in
LaTeX!
9033    }
9034    \listcsgadd{eled@sections#5@@}{#3}%
9035  }
9036
9037  %

```

End of the conditional test about noeledsec option.

```

9038  }{}
9039  %

```

### XXXIII 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.

```

9040 \def\normal@page@break{}
9041 %

```

**\l@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.

```

9042 \def\l@prev@pb{}
9043 \def\l@prev@nopb{}
9044 %

```

**\ledpb** The \ledpb macro writes the call to \led@pb in line-list file. The \ledpbnum macro writes the call to \led@pbnum in line-list file. The \lednopb macro writes the call to \led@nopb in line-list file. The \lednopbnum macro writes the call to \led@nopbnum in line-list file.

```

9045 \newcommand{\ledpb}{\write\linenum@out{\string\led@pb}}
9046 \newcommand{\ledpbnum}[1]{\write\linenum@out{\string\led@pbnum{#1}}}
9047 \newcommand{\lednopb}{\write\linenum@out{\string\led@nopb}}
9048 \newcommand{\lednopbnum}[1]{\write\linenum@out{\string\led@nopbnum{#1}}}
9049 %

```



`\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.

```

9050 \newcommand{\led@pb}{\listxadd{\l@prev@pb}{\the\absline@num}}
9051 \newcommand{\led@pbnum}[1]{\listxadd{\l@prev@pb}{#1}}
9052 \newcommand{\led@nopb}{\listxadd{\l@prev@nopb}{\the\absline@num}}
9053 \newcommand{\led@nopbnum}[1]{\listxadd{\l@prev@nopb}{#1}}
9054 %

```

`\ledpbsetting` The `\ledpbsetting` macro only changes the value of `\led@pb@macro`, for which the default value is before.

`\led@pb@setting`

```

9055 \def\led@pb@setting{before}
9056 \newcommand{\ledpbsetting}[1]{\gdef\led@pb@setting{#1}}
9057 %

```

`\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`

```

9058 \newcommand{\led@check@pb}{\xifinlist{\the\absline@num}{\l@prev@pb}{\pagebreak[4]}}
9059 \newcommand{\led@check@nopb}{%
9060   \IfStrEq{\led@pb@setting}{before}{%
9061     \xifinlist{\the\absline@num}{\l@prev@nopb}{%
9062       {\numdef{\abs@prevline}{\the\absline@num-1}%
9063       \xifinlist{\abs@prevline}{\normal@page@break}%
9064       {\nopagebreak[4]\enlargethispage{\baselineskip}}%
9065       {}}%
9066     {}}%
9067   }%
9068 }%
9069 \IfStrEq{\led@pb@setting}{after}{%
9070   \xifinlist{\the\absline@num}{\l@prev@nopb}{%
9071     \xifinlist{\the\absline@num}{\normal@page@break}%
9072     {\nopagebreak[4]\enlargethispage{\baselineskip}}%
9073     {}}%
9074 }%
9075 {}}%
9076 {}%
9077 {}%
9078 }
9079 %

```

## XXXIV 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.

```

9080 \newcommand{\check@pb@in@verse}{%
9081   \ifinstanza\iflednopbinverse\ifinserthangingsymbol% Using stanzas and
enabling page breaks in verse control, while on a hanging verse.
9082   \ifnum\page@num=\last@page@num\else%If we have change page
9083     \IfStrEq{\led@pb@setting}{before}{%
9084       \numgdef{\abs@line@verse}{\the\absline@num-1}%
9085       \ledpbnum{\abs@line@verse}%
9086     }{}%
9087     \IfStrEq{\led@pb@setting}{after}{%
9088       \numgdef{\abs@line@verse}{\the\absline@num-1}%
9089       \lednopbnum{\abs@line@verse}%
9090     }{}%
9091   \fi%
9092 \fi\fi\fi%
9093 }
9094 %

```

## XXXV Tools for hyperref package

`\Hy@raisedlink@left` The `hyperref` package provides a `\Hy@raisedlink` command, to be used to add an anchor to the top of a line and not to the bottom of it.<sup>35</sup>

However, this command disrupts the line breaking mechanism when it is called before any word. This is why `reledmac` defines `\Hy@raisedlink@left` that is called to the left of words, at the beginning of `\edtext` or inside the `\edlabel` commands.<sup>36</sup>

```

9095 \def\Hy@raisedlink@left#1{%
9096   \ifvmode
9097     #1%
9098   \else

```

<sup>35</sup><http://tex.stackexchange.com/a/17138/7712>.

<sup>36</sup>The code is inspired by an answer given by @unbonpetit. Thanks to him. <http://texnique.fr:80/osqa/questions/781/hyraisedlink-perturbe-la-maniere-dont-se-fait-la-coupure-de-ligne/801>.

```

9099 \Hy@SaveSpaceFactor
9100 \llap{\smash{%
9101 \begingroup
9102 \let\HyperRaiseLinkLength\@tempdima
9103 \setlength\HyperRaiseLinkLength\HyperRaiseLinkDefault
9104 \HyperRaiseLinkHook
9105 \expandafter\endgroup
9106 \expandafter\raise\the\HyperRaiseLinkLength\hbox{%
9107 \Hy@RestoreSpaceFactor
9108 #1%
9109 \Hy@SaveSpaceFactor
9110 }%
9111 }}%
9112 \Hy@RestoreSpaceFactor
9113 \penalty\@M\hskip\z@\relax
9114 \fi
9115 }
9116 %

```

## XXXVI Compatibility with eledmac

Here, we define some commands for the eledmac-compat option.

```

9117 \ifeledmaccompat{%
9118
9119 \newcommand{\footnormalX}[1]{\arrangementX[#1]{normal}}%
9120 \newcommand{\footparagraphX}[1]{\arrangementX[#1]{paragraph}}%
9121 \newcommand{\foottwocolX}[1]{\arrangementX[#1]{twocol}}%
9122 \newcommand{\footthreecolX}[1]{\XarrangementX[#1]{threecol}}%
9123
9124 \unless\ifnocritical@
9125 \newcommand{\footnormal}[1]{\Xarrangement[#1]{normal}}%
9126 \newcommand{\footparagraph}[1]{\Xarrangement[#1]{paragraph}}%
9127 \newcommand{\foottwocol}[1]{\Xarrangement[#1]{twocol}}%
9128 \newcommand{\footthreecol}[1]{\Xarrangement[#1]{threecol}}%
9129 \let\hsizetwocol\Xhsizetwocol
9130 \let\hsizethreecol\Xhsizethreecol
9131 \let\bhookXnote\Xbhooknote
9132 \let\boxsymlinenum\Xboxsymlinenum
9133 \let\symlinenum\Xsymlinenum
9134 \let\beforenumberinfootnote\Xbeforenumber
9135 \let\afternumberinfootnote\Xafternumber
9136 \let\beforeXsymlinenum\Xbeforesymlinenum
9137 \let\afterXsymlinenum\Xaftersymlinenum
9138 \let\inplaceofnumber\Xinplaceofnumber
9139 \let\Xlemmaseparator\lemmaseparator
9140 \let\afterlemmaseparator\Xafterlemmaseparator
9141 \let\beforelemmaseparator\Xbeforelemmaseparator
9142 \let\inplaceoflemmaseparator\Xinplaceoflemmaseparator

```

```

9143 \let\txtbeforeXnotes\Xtxtbeforenotes
9144 \let\afterXrule\Xafterrule
9145 \let\numberonlyfirstinline\Xnumberonlyfirstinline
9146 \let\numberonlyfirstintwlines\Xnumberonlyfirstintwlines
9147 \let\nonumberinfootnote\Xnonumberinfootnote
9148 \let\pstartinfootnote\Xpstart
9149 \let\pstartinfootnoteeverytime\Xpstarteverytime
9150 \let\onlyXpstart\Xonlypstart
9151 \let\Xnonumberinfootnote\Xnonumber
9152 \let\nonbreakableafternumber\Xnonbreakableafternumber
9153 \let\maxhXnotes\Xmaxhnotes
9154 \let\beforeXnotes\Xbeforenotes
9155 \let\boxlinenum\Xboxlinenum
9156 \let\boxlinenumalign\Xboxlinenumalign
9157 \let\boxstartlinenum\Xboxstartlinenum
9158 \let\boxendlinenum\Xboxendlinenum
9159 \let\twlines\Xtwlines
9160 \let\morethantwlines\Xmorethantwlines
9161 \let\twlinesbutnotmore\Xtwlinesbutnotmore
9162 \let\twlinesonlyinsamepage\Xtwlinesonlyinsamepage
9163 \fi
9164
9165 \unless\ifnofamiliar@
9166   \let\notesXwidthliketwocolumns\noteswidthliketwocolumnsX
9167 \fi
9168 \newcommandx{\parafootsep}[2][1,usedefault]{%
9169   \Xparafootsep[#1]{#2}%
9170   \parafootsepX[#1]{#2}
9171 }%
9172
9173 \newcommandx{\afternote}[2][1,usedefault]{%
9174   \Xafternote[#1]{#2}%
9175   \afternoteX[#1]{#2}%
9176 }%
9177
9178 \unless\ifnoend@
9179   \let\XendXtwlines\Xendtwlines
9180   \let\XendXmorethantwlines\Xendmorethantwlines
9181   \let\XhookXendnote\Xendhooknote
9182   \let\XboxXendlinenum\Xendboxlinenum%
9183   \let\XboxXendlinenumalign\Xendboxlinenumalign%
9184   \let\XboxXendstartlinenum\Xendboxstartlinenum%
9185   \let\XboxXendendlinenum\Xendboxendlinenum%
9186   \let\XendXlemmaseparator\Xendlemmaseparator
9187   \let\XendXbeforelemmaseparator\Xendbeforelemmaseparator
9188   \let\XendXafterlemmaseparator\Xendafterlemmaseparator
9189   \let\XendXinplaceoflemmaseparator\Xendinplaceoflemmaseparator
9190 \fi
9191
9192 \AtBeginDocument{%

```

```
9193 \ifdef\lineref{}\let\lineref\edlineref}%  
9194 }%  
9195  
9196  
9197 \fi%  
9198 %
```

</code>

## Appendix A Things to do when changing versions

### 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>37</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:

<pre>I saw my friend \critext{Smith} \Afootnote{Jones C, D.}/ on Tuesday.</pre>	<pre>1 I saw my friend 2 Smith on Tuesday. 2 Smith] Jones C, D.</pre>
---	---

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:

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

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:

```
\catcode`\<=\active
```

<sup>37</sup>A name like `\text` is likely to be defined by other  $\TeX$  packages (it certainly is by the AMS packages) and it seems sensible to try and avoid clashes with other definitions.

```
\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. 123 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 6.7.1 p. 35), 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 (7 p. 36) or `\Xfootnote` options (6.2.2 p. 25). Otherwise please add a new ticket on Github to request a new function for doing this.<sup>38</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>38</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. 9.3 p. 51). 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 5.2.3 p. 18). 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 16.2 p. 67.



## 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 19.1.3 p. 70 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 <code>\renewcommand</code></i>	<i>Replaced with</i>
<code>\@led@extranofeet</code>	<code>\newseries</code>
<code>\apprefprefixmore</code>	<code>\setapprefprefixmore</code>
<code>\apprefprefixsingle</code>	<code>\setapprefprefixsingle</code>
<code>\endstanzaextra</code>	Optional argument of <code>\&amp;</code>
<code>\hangingsymbol</code>	<code>\sethangingsymbol</code>
<code>\ledfootinsdim</code>	<code>\Xmaxhnotes</code> and <code>\maxhnotesX</code>
<code>\parafootftmsep</code>	<code>\Xparafootsep</code> and <code>\parafootsepX</code>
<code>\notenumfont</code>	<code>\Xnotenumfont</code> , <code>\Xendnotenumfont</code> and <code>\notenumfontX</code>
<code>\notefontsetup</code>	<code>\Xnotefontsize</code> , <code>\Xendnotefontsize</code> and <code>\notefontsizeX</code>
<code>\sidenoteseq</code>	<code>\setsidenotsep</code>
<code>\startstanzahook</code>	Optional argument of <code>\stanza</code>
<code>\symplinenum</code>	<code>\Xsymplinenum</code>

#### 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>
<code>\footparagraph</code>	<code>\Xarrangement</code>
<code>\footnormal</code>	<code>\Xarrangement</code>
<code>\foottwocol</code>	<code>\Xarrangement</code>
<code>\footthreecol</code>	<code>\Xarrangement</code>
<code>\footparagraphX</code>	<code>\arrangementX</code>
<code>\footnormalX</code>	<code>\arrangementX</code>
<code>\foottwocolX</code>	<code>\arrangementX</code>
<code>\footthreecolX</code>	<code>\arrangementX</code>
<code>\afterlemmaseparator</code>	<code>\Xafterlemmaseparator</code>
<code>\afternote</code>	<code>\Xafternote</code> and <code>\afternoteX</code>
<code>\afternumberinfootnote</code>	<code>\Xafternumber</code>
<code>\afterXrule</code>	<code>\Xafterrule</code>
<code>\afterXsymplinenum</code>	<code>\Xaftersymplinenum</code>
<code>\beforelemmaseparator</code>	<code>\Xbeforelemmaseparator</code>
<code>\beforenumberinfootnote</code>	<code>\Xbeforenumber</code>
<code>\beforeXnotes</code>	<code>\Xbeforenotes</code>
<code>\beforeXsymplinenum</code>	<code>\Xbeforesymplinenum</code>

<i>Old command</i>	<i>New command</i>
<code>\bhookXnote</code>	<code>\Xbhookendnote</code>
<code>\bhookXnote</code>	<code>\Xbhooknote</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.

## Appendix A.10 Migration to *reledmac* 2.1.0

*Reledmac* 2.1.0 fix some bugs when using `\Xbhooknote` and `\bhooknoteX` not in order to execute code at the beginning of each notes, but to insert content of at the beginning of each notes.

People who use these commands to do it, which is not the original idea, must change the following:

1. Horizontal space is no longer automatically added after the content of the `\Xbhooknote/\bhooknoteX` argument. You must include it manually. So instead of `\Xbhooknote{content}`, use `\Xbhooknote{content }.`
2. Indent is no longer automatically added before the content of the `\Xbhooknote/\bhooknoteX` argument. If you want to keep it, add `\indent` in the argument of `\Xbhooknote/\bhooknoteX`.

## Appendix A.11 Migration to *reledmac* 2.1.3

*Reledmac* 2.1.3 fix an historical bug, (style in *ledmac* 0.7!) which doubled the space before the rules of paragraphed familiar footnotes. Consequently, if you use paragraphed familiar footnotes, you should maybe adapt it, playing with `\beforenotesX`.

## Appendix A.12 Migration to *reledmac* 2.3.0

Before *reledmac* 2.3.0, for typesetting verse, any empty line was considered a paragraph inside verses. Counting empty lines this created breaking verse, hanging verses, and also added spurious vertical spaces. Version 2.3.0 disables paragraph in stanza. If you want vertical space, use optional argument of `\stanza` or `\endverse`.

### Appendix A.13 Migration to reledmac 2.4.0

It is not mandatory, but strongly recommended, to change any `\renewcommand{\endashchar}{\langle...}\rangle}` to the use of `\Xlinerangeseparator` or `/` and `\Xendlinerangeseparator` (7.2.4 p. 39).

### Appendix A.14 Migration to reledmac 2.5.0

It is strongly recommended to stop redefining `\printnpnum` and to use the hooks documented in 7.3 p. 42.

`\xlineref` does not print anymore the side flag (R for right side), because it is incompatible with numerical test. Use `\xflagref` to obtain it.

The `\printlines` and `\printendlines` commands take now an eighth argument, which is the side flag. It is strongly recommended to NEVER redefine these two commands and to use the setting commands instead (or to ask for new setting commands if the actual does not answer to your needs). However, if you have done it, just change your redefinition to have a new argument.

It is strongly recommended to stop redefining `\fullstop` and to use `\Xsublinesep` instead.

### Appendix A.15 Migration to reledmac 2.7.0

`\SErefonlypage` (introduced in reledmac 2.5.0) added an parenthesis after the page number. This was just an error, linked to a bad imitation of `\SErefwithpage`. That has been deleted. And so, the `\XendafterpagenumberSErefonlypage` to set it was also deleted.

`\rigidbalance` is split to two new commands: `\Xrigidbalance` for critical footnotes and `\rigidbalanceX` for familiar footnotes. If you have redefined it — but why should you have ? —, you should split your single redefinition in two redefinitions.

### Appendix A.16 Migration to reledmac 2.7.2

`\Xhsize` is already defined in the `floatrow` package. It becomes `\Xwidth`, and, consequently, `\hsizeX` becomes `\widthX`.

The ancient names are temporarily maintained as aliases.

### Appendix A.17 Migration to reledmac 2.8.0

Reledmac 2.8.0 fix spurious indents for paragraphed critical and familiar footnotes in `ledgroup` and `minipage`. You can re-establish the indent with `\Xparinden` and `\parindentX`.

### Appendix A.18 Migration to reledmac 2.13.1

Reledmac 2.5.0 added a bug, which makes the right flag to be printed on the right side of critical footnotes, even if not explicitly requested by using `\Xlineflag`.

Version 2.13.1 solves this issue. Please use `\Xlineflag` if you want to add the right flag.

### Appendix A.19 Migration to reledmac 2.18.0

After updating reledmac, and before any new compilation, you need to clean your `.aux` files, if you use `\edlabel` or related.

### Appendix A.20 Migration to reledmac 2.21.0

Previously, there were a bug, which was in contradiction with the handbook. If you wrote

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

“elephant” was indexed both in main text and in critical footnotes. With the new version of reledmac, it is indexed only in main text. If you want to index it also in critical footnotes, do

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

## 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 edstanzan.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 eledpar package*. December 2004. (Available from CTAN in `macros/latex/contrib/ledmmac`)

## Index

### Symbols

<code>\&amp;</code>	50
<code>\@EDROWFILL@</code>	1
<code>\@adv</code>	1
<code>\@advancestanzanumber</code>	1
<code>\@beforeinsertofthisedtext</code>	1
<code>\@doclearpage</code>	1
<code>\@doreinfeetX</code>	1
<code>\@edindex@fornote@</code>	1
<code>\@edindex@hyperref</code>	1
<code>\@edrowfill@</code>	1
<code>\@edtext@level</code>	1



\@emptytoks	1
\@fnpos	1
\@footnotemark	1
\@footnotetext	1
\@getfirstseries	1
\@gobblefour	1
\@gobbleseven	1
\@gobblethree	1
\@h	1
\@hangingsymbol	1
\@iiiminipage	1
\@insertstanza	1
\@k	1
\@l@dttempcnta	1
\@l@dttempcntb	1
\@lab	1
\@led@testifnofoot	1
\@lemma	1
\@line@num	1
\@lock	1
\@lopL	1
\@lopR	1
\@makechapterhead	1
\@makeschapterhead	1
\@mem@extranofeet	1
\@mem@old@ssect	1
\@mpfnpos	1
\@msd	1
\@msd@c	1
\@msd@options@iffullpage	1
\@msdata@list	1
\@nl	1
\@nl@reg	1
\@opXfeet	1
\@pend	1
\@pendR	1
\@ref	1
\@ref@reg	1
\@ref@reg@parse	1
\@sect	1
\@series	1
\@set	1
\@sidenotesep	1
\@ssect	1
\@startstanza	1
\@stopmsd	1
\@stopstanza	1
\@sw	1
\@tag	1
\@wredindex	1

\@xloop	1
\@xympar	1
CLASSarticle	70
CLASSbook	70, 342
CLASSmemoir	198, 258–260, 293, 342, 362, 421, 425
CLASSscrbook	425
COMMAND\*footnote	71
COMMAND\...@footnotemark...	201
COMMAND\...d@ta	149
COMMAND\<hook	
@<series	246
COMMAND\<hookname	
<pseudoseries	248, 249
COMMAND\<type	
footfmt	188
COMMAND\@@line	179
COMMAND\@MM	166, 422
COMMAND\@Rlineflag	295, 422
COMMAND\@Serefprefix	272
COMMAND\@Serefprefixmore	272
COMMAND\@add@	332
COMMAND\@adv	108
COMMAND\@apprefprefixmore	272
COMMAND\@apprefprefixsingle	272
COMMAND\@beforeinsertofthisedtext	130
COMMAND\@bsphack	261
COMMAND\@docclearpage	259, 260, 415, 425
COMMAND\@doreinfectX	425
COMMAND\@dprintingcolumns	422
COMMAND\@edindex@hyperref	295, 296
COMMAND\@edtext@level	127
COMMAND\@esphack	261
COMMAND\@fnpos	219, 255
COMMAND\@footnotemark	198, 199, 415, 425
COMMAND\@footnotetext	198, 199, 415
COMMAND\@gobble	125, 126, 241
COMMAND\@gobblefive	424
COMMAND\@gobblefour	421
COMMAND\@gobbleseven	243
COMMAND\@gobblethree	414
COMMAND\@h	182
COMMAND\@hangingsymbol	300
COMMAND\@iiiminipage	284, 286, 414, 425
COMMAND\@iiiminpage	284
COMMAND\@l	420
COMMAND\@l@dtmncnta	151, 153, 161
COMMAND\@l@dtmncntb	153
COMMAND\@l@reg	420
COMMAND\@lab	104, 261, 264, 268, 414
COMMAND\@ldunboxmpfoot	287

COMMAND\@led@extranofeet	363
COMMAND\@ledinnote@command	291
COMMAND\@lemma	130, 132
COMMAND\@lock	98, 300
COMMAND\@lopL	415
COMMAND\@lopR	415
COMMAND\@makecol	255, 256, 258, 425
COMMAND\@mpfnpos	219
COMMAND\@msd	309, 310
COMMAND\@msd@c	309
COMMAND\@msd@options@iffullpage	316
COMMAND\@msdata@list	310, 311
COMMAND\@nl	104–107, 109, 118, 264, 414, 415
COMMAND\@nl@reg	105, 360, 415, 420
COMMAND\@opXfeet	415
COMMAND\@opfeetX	425
COMMAND\@opxtrafeeti	425
COMMAND\@page	106, 264
COMMAND\@pend	415
COMMAND\@pendR	415
COMMAND\@ref	104, 114–116, 119, 120, 125
COMMAND\@ref@later	115, 121
COMMAND\@ref@reg	114, 415
COMMAND\@ref@reg@parsearg	115
COMMAND\@reinserts	255–258, 425
COMMAND\@secondoftwo	72
COMMAND\@sect	344
COMMAND\@series	245
COMMAND\@set	108
COMMAND\@sidenotesep	283
COMMAND\@stopmsd	310
COMMAND\@sw	116, 133, 136, 137
COMMAND\@tag	126, 128, 131
COMMAND\@tempcnta	84
COMMAND\@tempcntb	84
COMMAND\@toksa	90
COMMAND\@toksb	90
COMMAND\@xloop	162
COMMAND\@xympar	276, 425
COMMAND\Aendnote	16, 25
COMMAND\Afootfmt	165
COMMAND\Afootgroup	165
COMMAND\Afootnote	8, 16, 24, 25, 28, 128, 174, 198, 220, 236, 424
COMMAND\Afootstart	165
COMMAND\AtBeginDocument	258
COMMAND\AtEndEveryPend	19, 431
COMMAND\AtEveryPend	18, 19, 53, 143, 422, 423, 425, 431
COMMAND\AtEveryPend*	19
COMMAND\AtEveryPstart	18, 19, 53, 422, 423, 425, 428, 431
COMMAND\AtEveryPstart*	19

COMMAND\AtEveryStanza	53, 429, 431
COMMAND\AtEveryStopStanza	53, 429, 431
COMMAND\AtStartEveryPstart	19, 431
COMMAND\AtStartEveryStanza	53, 431
COMMAND\BeforeEveryStopStanza	431
COMMAND\Bendnote	16, 24
COMMAND\Bfootnote	8, 16, 198, 220, 236
COMMAND\Centering	46
COMMAND\Cfootnote	198
COMMAND\Columns	85, 171
COMMAND\Dfootnote	198
COMMAND\Efootnote	198
COMMAND\Gls	63
COMMAND\Hy@raisedlink	354
COMMAND\Hy@raisedlink@left	354
COMMAND\LTR	46
COMMAND\NR	344
COMMAND\Pages	85, 256, 257
COMMAND\ProcessOptionsX	75
COMMAND\RL	45
COMMAND\RaggedLeft	46
COMMAND\RaggedRight	46
COMMAND\SEonlypage	270, 427
COMMAND\SEref	57–59, 270, 273, 428, 430
COMMAND\SErefonlypage	57–59, 366, 427
COMMAND\SErefwithpage	57–59, 270, 273, 366, 427, 429
COMMAND\Stanza	420
COMMAND\Waklam	333
COMMAND\X@doreinfeet	257, 425
COMMAND\XR@prefix	276
COMMAND\XR@test	276
COMMAND\XR@test@mac	276
COMMAND\XR@test@mac@test	276
COMMAND\XXXXXXfmt	359
COMMAND\XXXXXXfmt	359
COMMAND\Xafterlemmaseparator	43, 363
COMMAND\Xafternote	46, 47, 362, 363
COMMAND\Xafternumber	41, 363
COMMAND\Xafterrule	48, 221, 363, 420, 423
COMMAND\Xaftersymlinenum	41, 363
COMMAND\Xarrangement	37, 47, 71, 166, 167, 247, 363
COMMAND\Xarrangement@footparagraph	172
COMMAND\Xarrangement@normal	167
COMMAND\Xarrangement@paragraph	172
COMMAND\Xbeforeinserting	45, 46
COMMAND\Xbeforelemmaseparator	43, 363
COMMAND\Xbeforenotes	47, 220, 363, 420, 423
COMMAND\Xbeforenumber	39, 41, 363
COMMAND\Xbeforesymlinenum	41, 363
COMMAND\Xbhookendnote	364

COMMAND\Xbhookgroup	47, 427, 428
COMMAND\Xbhooknote	45, 364, 365, 425, 426
COMMAND\Xboxendlinenum	42, 364, 424
COMMAND\Xboxlinenum	41, 42, 364
COMMAND\Xboxlinenumalign	42, 364, 424
COMMAND\Xboxstartlinenum	42, 364, 424
COMMAND\Xboxsymlinenum	41, 42, 364
COMMAND\Xcolalign	46, 423
COMMAND\Xdo@feet	425, 430
COMMAND\Xend	243
COMMAND\XendXafterlemmaseparator	364
COMMAND\XendXbeforelemmaseparator	364
COMMAND\XendXinplaceoflemmaseparator	364
COMMAND\XendXlemmaseparator	364
COMMAND\XendXmorethantwolines	364
COMMAND\XendXtwolines	364
COMMAND\Xendafterenumber	41, 426
COMMAND\Xendafterlemmaseparator	43, 364
COMMAND\Xendafternote	49, 428
COMMAND\Xendafternumber	43
COMMAND\Xendafterpagenumbe	429
COMMAND\Xendafterpagenumber	42, 59
COMMAND\XendafterpagenumberSErefonlypage	366
COMMAND\Xendaftersymlinenum	41, 43, 426
COMMAND\Xendahookinplaceofnumber	43, 426
COMMAND\Xendahooklinenum	43, 426
COMMAND\Xendbeforelemmaseparator	43, 364
COMMAND\Xendbeforelinenum	42
COMMAND\Xendbeforenumber	41, 426
COMMAND\Xendbeforepagenumber	42, 58, 59
COMMAND\XendbeforepagenumberSErefonlypage	58
COMMAND\Xendbeforesymlinenum	41, 43, 426
COMMAND\Xendbhookinplaceofnumber	43, 426
COMMAND\Xendbhooklinenum	42, 426
COMMAND\Xendbhooknote	45
COMMAND\Xendboxendlinenum	42, 424
COMMAND\Xendboxlinenum	42, 364, 422
COMMAND\Xendboxlinenumalign	42, 364, 424
COMMAND\Xendboxstartlinenum	42, 424
COMMAND\Xendboxsymlinenum	42, 426
COMMAND\Xendhangindent	45, 426, 428
COMMAND\Xendinplaceoflemmaseparator	26, 43, 364
COMMAND\Xendinplaceofnumber	41, 425
COMMAND\Xendinplaceofpagenumber	39, 430
COMMAND\Xendinsertsep@	227
COMMAND\Xendlemmadisablefontselection	44
COMMAND\Xendlemmafont	44, 427
COMMAND\Xendlemmaseparator	26, 43, 364
COMMAND\Xendletboxendlinenum	364
COMMAND\Xendlineflag	59

COMMAND\Xendlineprefixmore	42, 59
COMMAND\Xendlineprefixsingle	42, 59
COMMAND\Xendlinerangeseparator	39, 59, 186, 366, 426
COMMAND\Xendmorethantwolines	26, 40, 59, 364, 423, 424
COMMAND\Xendnonumber	40, 425
COMMAND\Xendnote	223, 242, 243, 423
COMMAND\Xendnotefontsize	44, 363
COMMAND\Xendnotenumfont	42–44, 363
COMMAND\Xendnotes	226
COMMAND\Xendnumberonlyfirstinline	38, 426
COMMAND\Xendnumberonlyfirstintwolines	38, 426
COMMAND\Xendpagenumberonlyfirst	38, 430
COMMAND\Xendpagenumberonlyfirstifsingle	38, 430
COMMAND\Xendpagenumberonlyfirstintwo	38, 430
COMMAND\Xendparagraph	49, 420
COMMAND\Xendsep	49
COMMAND\Xendsublinesep	40, 59, 186
COMMAND\Xendsymlinenum	38, 426
COMMAND\Xendsympagenum	39, 430
COMMAND\Xendtwolines	26, 39, 40, 59, 364, 423, 424
COMMAND\Xendtwolinesbutnotmore	40, 59, 423, 424
COMMAND\Xendtwolinesonlyinsamepage	40, 59, 423, 424
COMMAND\Xendwrapcontent	45, 429
COMMAND\Xendwraplemma	45, 429
COMMAND\Xfootfmt	365
COMMAND\Xfootgroup	171
COMMAND\Xfootins	170
COMMAND\Xfootnote	55, 61, 126, 359, 417, 421, 423, 427, 429
COMMAND\Xfootstarts	171
COMMAND\Xgroupbyline	47, 159, 196
COMMAND\Xgroupbylines	430
COMMAND\Xgroupbylineseparetwolines	47
COMMAND\Xhangindent	45, 426
COMMAND\Xhsize	366, 427, 428
COMMAND\Xhsizethreecol	46, 49, 364
COMMAND\Xhsizetwocol	46, 49, 248, 364
COMMAND\Xinplaceoflemmaseparator	25, 43, 364
COMMAND\Xinplaceofnumber	41, 364, 423, 424
COMMAND\Xinsertparafootsep	177, 178
COMMAND\Xledsetnormalparstuff	361, 362, 423
COMMAND\Xlemmadisablefontselection	44
COMMAND\Xlemmafont	44, 427
COMMAND\Xlemmaseparator	43, 186, 250, 252, 254, 364
COMMAND\Xlineflag	58, 366, 367, 429
COMMAND\Xlinerangeseparator	39, 58, 186, 366, 426
COMMAND\Xmaxhnotes	48, 70, 71, 363, 364, 420, 422
COMMAND\Xmorethantwolines	25, 39, 40, 58, 364, 422
COMMAND\Xnoindent	426
COMMAND\Xnolemmaseparator	43, 254, 364
COMMAND\Xnonbreakableafternumber	41, 364, 418

COMMAND\Xnonumber	40, 364
COMMAND\Xnonumberinfootnote	364
COMMAND\Xnotefontsize	44, 363
COMMAND\Xnotefontsize@⟨s⟩	177, 181, 182
COMMAND\Xnotenumfont	44, 363
COMMAND\Xnoteswidthliketwocolumns	49, 421
COMMAND\Xnumberonlyfirstinline	38, 39, 47, 101, 188, 249, 250, 252, 364, 417, 422, 431
COMMAND\Xnumberonlyfirstintwolines	38, 47, 364, 417, 431
COMMAND\Xonlypstart	40, 364, 417, 422
COMMAND\Xpagelinesep	41, 430
COMMAND\Xparafootsep	47, 101, 363, 364, 429, 431
COMMAND\Xparafootsep@series	177
COMMAND\Xparinden	366
COMMAND\Xparindent	45, 423, 426, 428
COMMAND\Xprenotes	48, 221, 429
COMMAND\Xprenotes@	170, 221, 417
COMMAND\Xpstart	40, 361, 364, 417, 422
COMMAND\Xpstarteverytime	40, 361, 364, 422
COMMAND\Xragged	47
COMMAND\Xrigidbalance	179, 366, 427
COMMAND\Xstanza	40, 53
COMMAND\Xstanzaseparator	40
COMMAND\Xstorelineinfo	188
COMMAND\Xsublinesep	22, 40, 41, 58, 186, 366
COMMAND\Xsublinesepside	22, 40
COMMAND\Xsymlinenum	38, 47, 363, 364, 424, 431
COMMAND\Xtoendnotes	27, 243
COMMAND\Xtwolines	25, 39, 40, 58, 193, 194, 248, 364, 422
COMMAND\Xtwolinesappref	248
COMMAND\Xtwolinesbutnotmore	39, 40, 58, 364, 423
COMMAND\Xtwolinesbutnotmoreappref	249
COMMAND\Xtwolinesonlyinsamepage	39, 40, 58, 364, 423
COMMAND\Xtxtbeforenotes	47, 364, 429, 430
COMMAND\Xunvxh	174, 365
COMMAND\Xwidth	49, 366, 428
COMMAND\Xwrapcontent	45, 429
COMMAND\Xwraplemma	44–46, 429
COMMAND\&	363
COMMAND\⟨XXX⟩vfootnote	196
COMMAND\absline@num	98, 150
COMMAND\accent	125
COMMAND\actionlines@list	99, 152
COMMAND\actions@list	99
COMMAND\add@Xgroupbyline	159
COMMAND\add@inserts	99, 158, 159
COMMAND\add@inserts@next	158, 159
COMMAND\add@msd@	309
COMMAND\add@msdata	310, 311
COMMAND\add@msdata@firstlineofpage	313
COMMAND\add@msddata	309

COMMAND\add@penalties	150, 160
COMMAND\addcontentsline	340
COMMAND\addfootins	359, 362
COMMAND\addfootinsX	359, 362
COMMAND\addtoendnotes	243
COMMAND\advancelabel@refs	263
COMMAND\advanceline	23, 100, 108, 122, 425
COMMAND\affixlin@num	283
COMMAND\affixline@num	153, 156, 157, 415
COMMAND\affixpstart@num	157
COMMAND\afterXrule	363
COMMAND\afterXsymlinenum	363
COMMAND\afterenumber	41
COMMAND\aftergroup	125, 129
COMMAND\afterlemmaseparator	363
COMMAND\afternote	363
COMMAND\afternoteX	47, 362, 363
COMMAND\afternumberinfootnote	363
COMMAND\afterruleX	48, 420, 423
COMMAND\applabel	57, 265, 266, 423, 429
COMMAND\appref	55, 57–59, 270, 273, 427, 428
COMMAND\apprefprefixmore	58, 363
COMMAND\apprefprefixsingle	58, 363
COMMAND\apprefwithpage	57–59, 270, 273, 424, 427
COMMAND\arrangementX	37, 71, 202, 247, 363
COMMAND\arrangementX@normal	207
COMMAND\article	15
COMMAND\at@every@pend	143
COMMAND\autopar	18, 140, 144, 145, 217, 416, 418, 419, 423
COMMAND\ballast	70
COMMAND\ballast@count	150, 161
COMMAND\baselineskip	37, 173, 177
COMMAND\beforeXnotes	363
COMMAND\beforeXsymlinenum	363
COMMAND\beforeelectedchapter	10, 68, 341
COMMAND\beforeinsertingX	45
COMMAND\beforelemmaseparator	363
COMMAND\beforenotesX	48, 365, 419, 420, 423
COMMAND\beforenumberinfootnote	363
COMMAND\begin	317
COMMAND\beginnumbering	16, 17, 19, 20, 85, 86, 88, 97, 102, 117, 144, 223, 309, 417, 420, 424, 425, 430, 431
COMMAND\bf	417
COMMAND\bfseries	44, 417
COMMAND\bhookXnote	364
COMMAND\bhookgroupX	47, 427
COMMAND\bhooknoteX	45, 365, 425, 426
COMMAND\body	301
COMMAND\bodyfootmarkA	34
COMMAND\book	15



COMMAND\boxXendlinenum	364
COMMAND\boxXendlinenumalign	364
COMMAND\boxXendstartlinenum	364
COMMAND\boxendlinenum	364
COMMAND\boxlinefootnote	190
COMMAND\boxlinenum	364
COMMAND\boxlinenumalign	364
COMMAND\boxstartlinenum	364
COMMAND\boxsymlinenum	364
COMMAND\break	37, 175
COMMAND\brokenpenalty	160
COMMAND\centering	46
COMMAND\ch@ck@l@ck	415
COMMAND\ch@cksub@l@ck	156, 415
COMMAND\chapter	67, 342, 346, 420, 423, 425, 429
COMMAND\chaptermark	340
COMMAND\check@pb@in@verse	354
COMMAND\colalignX	46, 423
COMMAND\collect@body	318
COMMAND\color	429
COMMAND\colorbox	71
COMMAND\columns	49
COMMAND\columnwidth	173, 421
COMMAND\command names	248, 249
COMMAND\copyright	126
COMMAND\correct@Xfootins@box	422
COMMAND\correct@footinsX@box	422
COMMAND\count	180
COMMAND\critex	416
COMMAND\critext	132, 358, 359, 362
COMMAND\curname	76, 135
COMMAND\csquotes	240
COMMAND\ctab	333, 338
COMMAND\ctabtext	338
COMMAND\dc col	327
COMMAND\def	73
COMMAND\detokenize	135
COMMAND\dimen	180
COMMAND\dimexpr	49
COMMAND\discretionary	174
COMMAND\displaywidowpenalty	160
COMMAND\do@Xfeet	256, 415, 425, 430
COMMAND\do@actions	150–152, 415
COMMAND\do@actions@fixedcode	415
COMMAND\do@actions@next	151, 152
COMMAND\do@ballast	150, 161
COMMAND\do@feet@custom@order	255
COMMAND\do@insidelinehook	418
COMMAND\do@line	99, 124, 142, 145, 149, 158, 160, 300, 415, 416, 418, 420
COMMAND\do@linehook	415

COMMAND\do@lockoff	100
COMMAND\do@lockon	100
COMMAND\dodoreintrafeet	414
COMMAND\doendnotes	26, 227, 365, 424, 431
COMMAND\doendnotesbysection	26, 227, 243, 424, 431
COMMAND\doennotes	431
COMMAND\doinsidelinehook	24, 421
COMMAND\dolinehook	24, 421
COMMAND\doreintrafeeti	425
COMMAND\doreintrafeetii	425
COMMAND\doxtrafeet	255, 414
COMMAND\doxtrafeeti	425
COMMAND\doxtrafeetii	425
COMMAND\dummy@ref	125
COMMAND\edaftertab	66, 333, 334
COMMAND\edatleft	66, 331
COMMAND\edatright	66, 331
COMMAND\edbeforetab	66, 333
COMMAND\edfilldimen	332
COMMAND\edfont@info	131
COMMAND\edgls	63, 290
COMMAND\edglsadd	431
COMMAND\edgls...	428
COMMAND\edindex	61–63, 289, 290, 294, 296, 322, 418, 421, 422, 425, 426, 430–432
COMMAND\edindexlab	63
COMMAND\edlabel	55, 57, 59, 126, 261, 263, 264, 267, 268, 275, 290, 322, 354, 367, 414, 417–419, 422, 427
COMMAND\edlabelE	57, 266
COMMAND\edlabelS	57, 266
COMMAND\edlabelSE	57
COMMAND\edlineref	55, 261, 364, 422, 424, 427, 431
COMMAND\edmakelabel	57, 275
COMMAND\edpageref	55, 261, 267, 275
COMMAND\edrowfill	333
COMMAND\edsublineref	55
COMMAND\edtabcolsep	326
COMMAND\edtext	6, 24, 25, 27–32, 50, 55–57, 61, 64, 71, 99, 114, 116, 119, 120, 123–132, 134–136, 138, 265, 266, 269, 321, 323, 342, 354, 358, 359, 362, 414, 416, 418, 420–424, 429–432
COMMAND\edtext@level	424
COMMAND\edtextlater	116
COMMAND\edvertdots	67, 332
COMMAND\edvertline	66, 67, 331
COMMAND\elechapter	68
COMMAND\eled@sectioning@out	347
COMMAND\eledchapter	67, 362, 421, 425
COMMAND\eledchapter*	67
COMMAND\eledmac@error	414
COMMAND\eledsection	7, 16, 67, 125, 148, 341, 362, 423, 431
COMMAND\eledsection*	67
COMMAND\eledsubsection	67, 362

COMMAND\eledsubsection*	67
COMMAND\eledsubsubsection	67, 362
COMMAND\eledsubsubsection*	67
COMMAND\eledxxx	10, 68, 347, 420
COMMAND\eledxxxx	340
COMMAND\elesection	347
COMMAND\else	289, 341
COMMAND\empty	84, 154, 261
COMMAND\end	317
COMMAND\end@lemmas	125
COMMAND\endashchar	186
COMMAND\endgraf	142, 176, 217
COMMAND\endlock	22, 100, 123, 305
COMMAND\endminipage	284, 286, 414, 425
COMMAND\endmsdata	33
COMMAND\endnotes	423, 427
COMMAND\endnumbering	16, 17, 20, 85, 86, 88, 415, 424, 430, 431
COMMAND\endprint	223, 226, 243, 361
COMMAND\endstanzaextra	363
COMMAND\endsub	22, 100, 121
COMMAND\endverse	365
COMMAND\everypar	144
COMMAND\extensionchars	69, 85
COMMAND\externaldocument	59, 275
COMMAND\f@x@l@cks	415
COMMAND>falseverse	362, 418, 420
COMMAND\fi	341
COMMAND\firstlinenum	21, 153, 416
COMMAND\firstsublinenum	21, 416
COMMAND\fix@page	105, 106, 415
COMMAND\flag@end	119, 120, 131, 420
COMMAND\flag@end@RTL	120
COMMAND\flag@end@later	121
COMMAND\flag@start	119, 120, 131, 420, 421
COMMAND\flag@start@RTL	120
COMMAND\flag@start@later	121
COMMAND\flagstanza	54
COMMAND\floatingpenalty	166, 422
COMMAND\flush@notes	161, 162
COMMAND\fnpos	36, 219, 419, 430
COMMAND\footfmt	165, 168
COMMAND\footfmt...	202
COMMAND\footfootmarkA	34
COMMAND\footfudgefactor	175
COMMAND\footfudgefiddle	71, 173, 414
COMMAND\footgroup	165
COMMAND\footins	170
COMMAND\footnormal	248, 363, 414
COMMAND\footnormalX	363
COMMAND\footnote	34, 70, 198, 199, 360, 415

COMMAND\footnote@lang	186
COMMAND\footnoteA	16, 34
COMMAND\footnoteB	16
COMMAND\footnoteC	24
COMMAND\footnoteE	34
COMMAND\footnoteX	8, 35, 240, 241, 429
COMMAND\footnoteX@reading	241
COMMAND\footnoteXmark	241, 431
COMMAND\footnoteXmk	254
COMMAND\footnoteXtext	241, 431
COMMAND\footnote⟨X⟩	126
COMMAND\footnote⟨X⟩mark	35
COMMAND\footnote⟨X⟩mk	35
COMMAND\footnote⟨X⟩nomk	35
COMMAND\footnote⟨X⟩text	35
COMMAND\footnotelang@lua	164
COMMAND\footnotelang@poly	164
COMMAND\footnotemark	35, 241
COMMAND\footnoteoption@	163, 426
COMMAND\footnoterule	180
COMMAND\footnotesize	44
COMMAND\footnotetext	35, 241
COMMAND\footparagraph	173, 248, 363, 420
COMMAND\footparagraphX	212, 363, 420
COMMAND\footplitskips	415, 422
COMMAND\footstart	165, 170, 180
COMMAND\footstrut	176
COMMAND\footthreecol	363
COMMAND\footthreecolX	363, 424
COMMAND\foottwocol	363
COMMAND\foottwocolX	363, 424
COMMAND\foreignlanguage	45
COMMAND\fullstop	366
COMMAND\get@edindex@hyperref	295
COMMAND\get@edindex@ledinnote@command	291
COMMAND\get@fnmark	199
COMMAND\get@index@command	419
COMMAND\get@linelistfile	415
COMMAND\get@thisfootnote	206
COMMAND\getline@num	150, 151
COMMAND\gl@p	90
COMMAND\global	104
COMMAND\globaldefs	104
COMMAND\gls	63, 298
COMMAND\hangindentX	45, 423, 426
COMMAND\hangingsymbol	363, 416
COMMAND\hbox	174
COMMAND\hfill	419
COMMAND\hidenumbering	23, 113, 423
COMMAND\hidenumberingonleftpage	23, 113, 429

COMMAND\hidenumberingonrightpage	23, 113, 429
COMMAND\hline	64
COMMAND\hrulefill	333
COMMAND\hsize	38, 170, 173, 175, 181, 184, 218, 415, 421
COMMAND\hsizeX	366, 427, 428
COMMAND\hsizethreecol	364
COMMAND\hsizethreecolX	46, 49
COMMAND\hsizetwocol	364
COMMAND\hsizetwocolX	46, 49
COMMAND\hyperlinkR	295
COMMAND\hyperlinkformat	294
COMMAND\hyperlinkformatR	295
COMMAND\if@RTL	77, 129
COMMAND\if@edtext@	421, 424
COMMAND\if@eled@sectioning	347
COMMAND\if@firstlineofpage	77
COMMAND\if@firstlineofpageR	77
COMMAND\if@msd@options@fullpage	316
COMMAND\if@msdata@insertedfrompreviouspage	313
COMMAND\if@nobreak	143
COMMAND\if@noneed@Footnote	119
COMMAND\ifXnote@	85
COMMAND\ifbypage@	91
COMMAND\ifbypage@R	91
COMMAND\ifbypstart@	91
COMMAND\ifbypstart@R	91
COMMAND\iffirst@linenum@out@	117, 118
COMMAND\ifindtl@innote	85
COMMAND\ifindtl@notenumber	85
COMMAND\ifinserthangingsymbol	300
COMMAND\ifinstanza	300
COMMAND\ifstwofollowinglines	194
COMMAND\ifl@d@Xmorethantwolines	191, 422
COMMAND\ifl@d@Xtwolines	191
COMMAND\ifl@d@dash	191
COMMAND\ifl@d@elin	191
COMMAND\ifl@d@esl	191
COMMAND\ifl@d@pnum	191
COMMAND\ifl@d@ssub	191
COMMAND\ifl@dend@X	242
COMMAND\ifl@dmemoir	414
COMMAND\ifl@dpaging	421
COMMAND\ifl@dpairing	84, 416
COMMAND\ifl@dprintingpages	422
COMMAND\ifl@dskipnumber	153
COMMAND\ifl@dstartendok	332
COMMAND\ifl@imakeidx	77
COMMAND\ifledRcol	85, 416
COMMAND\ifledRcol@	85, 420
COMMAND\iflemmacommand@	421

COMMAND\ifnoend@	228
COMMAND\ifnoledgroup@	289
COMMAND\ifnoteschanged@	101
COMMAND\ifnumberedpar@	139
COMMAND\ifnumbering	86, 88
COMMAND\ifnumberingR	85, 416
COMMAND\ifnumberline	130, 153
COMMAND\ifpst@rted	416
COMMAND\ifpst@rtedL	86
COMMAND\ifseriesbefore	246
COMMAND\ifstopmsdata@inserted@	309
COMMAND\ifsublines@	98, 110
COMMAND\iftrue	424
COMMAND\ifvmode	263
COMMAND\ifxxx	341
COMMAND\ignorespaces	128
COMMAND\imki@wrindexentry	77
COMMAND\immediate	117, 118, 222
COMMAND\indent	18, 144, 365
COMMAND\index	297, 430
COMMAND\indtl@wrindexentry	77
COMMAND\initnumbering@quote	339, 425
COMMAND\initnumbering@reg	415
COMMAND\initnumbering@sectcmd	425
COMMAND\inplaceoflemmaseparator	364
COMMAND\inplaceofnumber	364
COMMAND\insert	130, 158, 165, 168, 182, 196, 202
COMMAND\insert@Xtxtbeforenotes	162, 182
COMMAND\insert@count	114, 119, 128
COMMAND\insert@countR	128
COMMAND\insert@msdata	310, 316
COMMAND\inserthangingsymbol	419
COMMAND\insertlines@list	99, 114
COMMAND\insertparafootsepX	216
COMMAND\inserts@list	124, 140, 158, 159, 174
COMMAND\interAfootnotelinepenalty	416
COMMAND\interfootnotelinepenalty	416
COMMAND\interlinepenalty	165
COMMAND\interparanoteglue	362
COMMAND\justifying	46
COMMAND\l@advance@parledegroupp@beforenormalnotes	425
COMMAND\l@d@@wrindexhyp	421
COMMAND\l@d@add	133
COMMAND\l@d@end	223, 242
COMMAND\l@d@nums	128, 131–133, 191
COMMAND\l@d@section	223
COMMAND\l@d@set	109, 122
COMMAND\l@dampcount	323
COMMAND\l@dbfnote	199, 415
COMMAND\l@dcheckstartend	332

COMMAND\l@dchset@num	109
COMMAND\l@dcolcount	323, 324
COMMAND\l@dcollect@@body	317
COMMAND\l@dcollect@body	317
COMMAND\l@dcsnote	420
COMMAND\l@dcsnotetext	149, 280
COMMAND\l@dcsnotetext@l	149, 280
COMMAND\l@dcsnotetext@r	149, 280
COMMAND\l@ddodoreintrafeet	256, 414
COMMAND\l@ddoxtrafeet	256, 414
COMMAND\l@demptyd@ta	416
COMMAND\l@dend@close	222
COMMAND\l@dend@open	222
COMMAND\l@dend@stuff	223
COMMAND\l@denvbody	317
COMMAND\l@dfeetbeginmini	414
COMMAND\l@dfeetendmini	414
COMMAND\l@dgetline@margin	416
COMMAND\l@dgetlock@disp	416
COMMAND\l@dgetref@num	268, 269
COMMAND\l@dgetsidenote@margin	277, 416
COMMAND\l@dgobbeloptarg	421
COMMAND\l@dgobblearg	421
COMMAND\l@dgobbleoptarg	322
COMMAND\l@dlabel@parse	268, 269
COMMAND\l@dld@ta	153, 155
COMMAND\l@dlp@rbox	282
COMMAND\l@dlsn@te	416
COMMAND\l@dlsnote	420
COMMAND\l@dmake@labels	263, 264, 276
COMMAND\l@dmake@labelsR	276
COMMAND\l@dnumpstartsL	86, 416
COMMAND\l@dp@rsefootspec	191
COMMAND\l@dparsefootspec	191
COMMAND\l@dpush@begins	317
COMMAND\l@drd@ta	153, 155
COMMAND\l@dref@undefined	268
COMMAND\l@drrsn@te	416
COMMAND\l@drrsnote	420
COMMAND\l@dtabaddcols	332
COMMAND\l@dtabnoexpands	414
COMMAND\l@dumboxmpfoot	425
COMMAND\l@dunboxmpfoot	416
COMMAND\l@dzeropenalties	416, 421
COMMAND\l@pb	353
COMMAND\l@prev@nopb	352
COMMAND\l@prev@pb	352
COMMAND\l@reg	360
COMMAND\label	19, 57, 59, 63, 261, 262, 269
COMMAND\label@refs	261

COMMAND\labelstarttrue	19, 417
COMMAND\labelref@list	261, 264
COMMAND\language	174
COMMAND\last@page@num	415
COMMAND\lastbox	144
COMMAND\lastskip	121
COMMAND\latex@makecol	258
COMMAND\leavevmode	18, 144
COMMAND\led@check@nopb	353
COMMAND\led@check@pb	353
COMMAND\led@nopb	352–354
COMMAND\led@nopbnum	352, 353
COMMAND\led@pb	352–354
COMMAND\led@pb@macro	353
COMMAND\led@pbnum	352, 353
COMMAND\led@reinit@index@fornote	297
COMMAND\led@set@index@fornote	297
COMMAND\ledRflag	295
COMMAND\ledchapter	362, 418
COMMAND\ledfootinsdim	363
COMMAND\ledinnernote	59, 278, 420, 429
COMMAND\ledinnote	292, 361, 424
COMMAND\ledinnotemark	61, 361, 423
COMMAND\ledleftnote	60, 278
COMMAND\ledlinenum	96, 416
COMMAND\ledllfill	149
COMMAND\ledlsnotefontsetup	429
COMMAND\ledlsnotesep	60
COMMAND\ledlsnotewidth	60
COMMAND\lednopb	68, 69, 352
COMMAND\lednopbinverse	354
COMMAND\lednopbinversetrue	52, 69
COMMAND\lednopbnum	352
COMMAND\ledouternote	60, 278, 420, 429
COMMAND\ledpb	68, 69, 352, 362
COMMAND\ledpbnum	352
COMMAND\ledpbsetting	69, 353, 426
COMMAND\ledrightnote	60, 278
COMMAND\ledrsnotefontsetup	429
COMMAND\ledrsnotesep	60
COMMAND\ledrsnotewidth	60
COMMAND\ledsection	362
COMMAND\ledsectnomark	340
COMMAND\ledsectnotoc	340
COMMAND\ledsetnormalparstuff	361, 362, 423
COMMAND\ledsetnormalparstuff@common	217
COMMAND\ledsetnormalparstuffX	361, 362, 423
COMMAND\ledsidenote	60, 278, 280
COMMAND\ledsubsection	362
COMMAND\ledsubsubsection	362



COMMAND\ledxxx	420
COMMAND\left	66
COMMAND\leftctab	333
COMMAND\leftheadline	96
COMMAND\leftlinenum	22, 96, 414, 416
COMMAND\leftltab	333
COMMAND\leftnoteupfalse	60
COMMAND\leftstartnum	157
COMMAND\leftftab	333
COMMAND\leftsidenote	280
COMMAND\leftskip	170, 173, 175
COMMAND\lemma	3, 25, 27–32, 124, 127–129, 131, 132, 134, 358, 416, 417, 424, 425, 427
COMMAND\lemmaseparator	364
COMMAND\let	28, 50, 126, 130, 241, 258, 305, 414
COMMAND\letboxXendendlinenum	364
COMMAND\line	179, 182
COMMAND\line@list	99, 115, 131
COMMAND\line@list@stuff	86, 102, 117, 414, 416
COMMAND\line@list@version	104
COMMAND\line@margin	92, 155, 277
COMMAND\line@num	97, 98, 100, 153, 414
COMMAND\line@set	132, 133
COMMAND\lineation	21, 91
COMMAND\linebreak	37
COMMAND\linenum	25, 27, 28, 56, 57, 124, 132, 267, 269, 275, 358, 428
COMMAND\linenum@out	117, 261, 264
COMMAND\linenumlist	21, 84, 154, 414
COMMAND\linenumstyle	23, 96, 414, 431
COMMAND\linenumincrement	21, 416
COMMAND\linenummargin	21, 92, 277
COMMAND\linenumr@p	96, 414, 416
COMMAND\linenumrep	96, 416
COMMAND\linenumsep	22, 60, 96, 278
COMMAND\linerrangesep@	254
COMMAND\lineref	261, 267, 275, 364, 422
COMMAND\list@clear	90
COMMAND\list@clearing@reg	416
COMMAND\list@create	90
COMMAND\lock@disp	95
COMMAND\lock@off	111
COMMAND\lock@on	110
COMMAND\lockdisp	22, 95
COMMAND\loop	162, 301
COMMAND\ltab	333, 334, 338
COMMAND\ltabtext	338
COMMAND\m@mmf@prepare	199
COMMAND\makeatletter	149
COMMAND\makehboxofhboxes	175, 177
COMMAND\makeindex	61, 294
COMMAND\makelabel	275

COMMAND\managestanza@modulo	302
COMMAND\marginpar	59, 70, 276, 277, 415
COMMAND\marginparwidth	60, 278
COMMAND\markboth	149
COMMAND\mathchardef	301
COMMAND\maxhXnotes	364
COMMAND\maxhnotesX	48, 71, 363, 419, 420, 422–424
COMMAND\maxlinesinpar@list	102
COMMAND\measurebody	335
COMMAND\measuretbody	335
COMMAND\memorybreak	20
COMMAND\morenoexpands	71, 72, 124, 126
COMMAND\morethantwolines	364
COMMAND\mpfnpos	36, 219, 419, 430
COMMAND\mpnormalfootgroup	415
COMMAND\mpnormalvfootnote	415
COMMAND\msdata	33, 307, 309, 310, 431
COMMAND\msdataposition	34
COMMAND\multfootsep	34, 198
COMMAND\multiplefootnotemarker	198
COMMAND\musixtex	420
COMMAND\n@num	416, 423
COMMAND\n@num@ref	423
COMMAND\new@line	118, 415
COMMAND\new@series	126
COMMAND\newcommand	28, 73, 198, 264
COMMAND\newcommandx	28, 29
COMMAND\newhookarg@specific	254
COMMAND\newhookcommand@series	248, 249, 423
COMMAND\newhookcommand@series@reload	249
COMMAND\newhookcommand@toggle@reload	249, 421
COMMAND\newhooktoggle@series	249, 423
COMMAND\newhooktoggle@specific	254
COMMAND\newif	423
COMMAND\newline	37
COMMAND\newlinechar	242
COMMAND\newseries	35, 359, 362, 363
COMMAND\newseries@	233, 234, 247
COMMAND\newverse	53, 54, 362, 420, 431
COMMAND\next	301
COMMAND\next@action	103
COMMAND\next@actionline	103
COMMAND\next@insert	159
COMMAND\nl@regR	105
COMMAND\no@expands	71, 126, 131, 241, 414
COMMAND\noXlemmaseparator	364
COMMAND\nobreak	191
COMMAND\nocritical	234
COMMAND\noeledsec	68, 362
COMMAND\noendnotes	362

COMMAND\noexpand	360
COMMAND\nofamiliar	251
COMMAND\noindent	18, 19, 52, 53, 144, 426
COMMAND\noindentX	426
COMMAND\nomk@	254
COMMAND\nonbreakableafternumber	364
COMMAND\nonnumberinfootnote	364
COMMAND\norelax	50
COMMAND\normal@footnotemarkX	202
COMMAND\normal@page@break	352
COMMAND\normal@pars	217
COMMAND\normalbfnoteX	416
COMMAND\normalbodyfootmarkX	202
COMMAND\normalfont	430
COMMAND\normalfootfmt	50, 169, 176, 186, 223, 428
COMMAND\normalfootfmtX	203
COMMAND\normalfootfootmarkX	203
COMMAND\normalfootgroup	171
COMMAND\normalfootgroupX	204
COMMAND\normalfootnoterule	166
COMMAND\normalfootstart	170, 173
COMMAND\normalfootstartX	204
COMMAND\normalvfootnote	168, 169
COMMAND\normalvfootnote@inserted	168, 169
COMMAND\normalvfootnoteX	202
COMMAND\notbool	341
COMMAND\notfontsetup	363
COMMAND\notfontsizeX	44, 363
COMMAND\notenumfont	363
COMMAND\notenumfontX	44, 363
COMMAND\notesXwidthliketwocolumns	364
COMMAND\noteswidthliketwocolumnsX	49, 364, 421, 423
COMMAND\num@lines	139, 161
COMMAND\numberlinefalse	20
COMMAND\numberlinetrue	20
COMMAND\numberonlyfirstinline	246, 364
COMMAND\numberonlyfirstintwolines	364
COMMAND\numberpstartfalse	19
COMMAND\numberpstarttrue	19, 40, 361, 417, 425, 431
COMMAND\numberstanza	40
COMMAND\numberstanzafalse	53
COMMAND\numberstanzatrue	53
COMMAND\numlabfont	22, 49, 96
COMMAND\one@line	139
COMMAND\onehalfspacing	426
COMMAND\onlyXpstart	364
COMMAND\onlysideX	240
COMMAND\page@action	100, 109
COMMAND\page@start	100, 416
COMMAND\pagecontents	100

COMMAND\pageparbreak	362
COMMAND\pageref	57, 267
COMMAND\par	27, 37, 144, 217
COMMAND\par@line	139, 161
COMMAND\para@footgroup	173
COMMAND\para@footgroupX	215
COMMAND\para@footsetup	173, 414
COMMAND\para@footsetupX	213, 414, 421
COMMAND\para@vfootnoteX	214
COMMAND\parafootfmt	175, 176, 428
COMMAND\parafootfmtX	215
COMMAND\parafootftm	178
COMMAND\parafootftmX	216
COMMAND\parafootftmsep	363
COMMAND\parafootsep	364, 419, 424
COMMAND\parafootsepX	47, 101, 363, 364, 429
COMMAND\parafootstart	173
COMMAND\parafootstartX	213
COMMAND\paravfootnote	174, 177
COMMAND\parfillskip	176
COMMAND\parindent	426
COMMAND\parindentX	45, 366, 426, 428
COMMAND\parshape	70
COMMAND\parskip	19, 53, 144
COMMAND\pausenumbering	20, 88, 89, 102, 104, 145, 419, 421, 428
COMMAND\penalty	176
COMMAND\pend	2, 7, 17–21, 23, 67, 68, 122, 124, 127, 133, 139, 140, 142–145, 157, 158, 360, 419, 420, 429–431
COMMAND\preXnotes	423, 429
COMMAND\prenotesX	48, 222, 423
COMMAND\prepare@Xgroupbyline	196
COMMAND\prepare@Xprenotes	220
COMMAND\prev@nopb	353
COMMAND\prev@pb	353
COMMAND\prevlineX	101
COMMAND\prevpageX@num	101
COMMAND\print@Xfootnoterule	423
COMMAND\print@Xnotes	256, 257
COMMAND\print@Xnotes@forpages	422
COMMAND\print@eledsection	148
COMMAND\print@footnoteXrule	423
COMMAND\print@leftmargin@eledsection	341
COMMAND\print@lemma	187
COMMAND\print@line	146
COMMAND\print@notesX@forpages	422
COMMAND\print@rightmargin@eledsection	341
COMMAND\printendlines	228, 233, 273, 366, 414, 416
COMMAND\printlinefootnote	188, 189, 422
COMMAND\printlinefootnotearea	190, 191, 422
COMMAND\printlinefootnotenumbers	188

COMMAND\printlines . . . . . 169, 186, 191, 192, 228, 273, 366, 414, 416, 422, 427  
 COMMAND\printnpnum . . . . . 361, 366  
 COMMAND\printpstart . . . . . 187  
 COMMAND\protect . . . . . 126, 360  
 COMMAND\providecommand . . . . . 198, 414  
 COMMAND\pstart . . . . . 2, 7, 17–21, 23, 67, 68, 109,  
 122, 127, 133, 139, 140, 142–144, 148, 158, 346, 360, 416–418, 420, 421, 423–425, 428–431  
 COMMAND\pstartinfootnote . . . . . 364  
 COMMAND\pstartinfootnoteeverytime . . . . . 364  
 COMMAND\pstartnum . . . . . 157  
 COMMAND\pstartref . . . . . 55, 261, 268, 419, 431  
 COMMAND\pstarts . . . . . 417  
 COMMAND\raggedX . . . . . 47  
 COMMAND\raggedleft . . . . . 46  
 COMMAND\raggedright . . . . . 46  
 COMMAND\raw@text . . . . . 139, 140  
 COMMAND\rbracket . . . . . 43  
 COMMAND\read@linelist . . . . . 102–104  
 COMMAND\ref . . . . . 57, 59, 63  
 COMMAND\reformatted@ . . . . . 273  
 COMMAND\reformattedwithpage . . . . . 273  
 COMMAND\relax . . . . . 19, 109, 130, 151, 159, 305, 322, 360  
 COMMAND\renewcommand . . . . . 71, 363, 366  
 COMMAND\reset@msd@options@iffullpage . . . . . 316  
 COMMAND\resetprevline@ . . . . . 101  
 COMMAND\resetprevpage@ . . . . . 101  
 COMMAND\resumenumbering . . . . . 20, 85, 88, 89, 102, 104, 145, 416, 420, 421, 428  
 COMMAND\right . . . . . 66  
 COMMAND\rightctab . . . . . 333  
 COMMAND\rightlinenum . . . . . 22, 96, 414, 416  
 COMMAND\rightltab . . . . . 334  
 COMMAND\rightnoteupfalse . . . . . 60  
 COMMAND\righttrtab . . . . . 334  
 COMMAND\rightsidenote . . . . . 280  
 COMMAND\rightright . . . . . 170, 173, 175, 176  
 COMMAND\rightstartnum . . . . . 157  
 COMMAND\rigidbalance . . . . . 179, 180, 182, 366, 427  
 COMMAND\rigidbalanceX . . . . . 179, 366, 427  
 COMMAND\robustify . . . . . 38  
 COMMAND\roman . . . . . 327, 427  
 COMMAND\rtab . . . . . 333, 334, 338  
 COMMAND\rtabtext . . . . . 335, 338  
 COMMAND\sameword . . . . . 29–32, 116, 133–135, 138, 422, 424, 426, 429, 431  
 COMMAND\sameword@inedtext . . . . . 134, 135  
 COMMAND\saweword . . . . . 134  
 COMMAND\scriptsize . . . . . 96  
 COMMAND\section . . . . . 67, 416, 430  
 COMMAND\section@num . . . . . 85  
 COMMAND\sectionmark . . . . . 340  
 COMMAND\select@lemmafont . . . . . 49, 50, 163

COMMAND\series	233, 234
COMMAND\series@	233
COMMAND\seriesatbegin	35, 245, 423
COMMAND\seriesatend	35, 246, 423
COMMAND\set@Xtxtbeforenotes	162
COMMAND\set@line	131
COMMAND\set@line@action	100, 110
COMMAND\setSErefonlypageprefixmore	58, 272, 428, 431
COMMAND\setSErefonlypageprefixsingle	58, 272, 428, 431
COMMAND\setSErefprefixmore	58
COMMAND\setSErefprefixsingle	58
COMMAND\setapprefprefixmore	58, 363
COMMAND\setapprefprefixsingle	58, 363, 427
COMMAND\setcommand@series	247
COMMAND\sethangingsymbol	52, 300, 363, 426
COMMAND\sethanginsymbol	50
COMMAND\setistwofollowinglines	194
COMMAND\setl@dlprbox	282
COMMAND\setline	23, 100, 105, 108, 122, 126, 142, 425
COMMAND\setlinenum	23, 105, 109, 122, 414
COMMAND\setmsdatalabel	34
COMMAND\setmsdataposition	431
COMMAND\setmsdataseries	34
COMMAND\setprintendlines	228, 230, 416
COMMAND\setprintlines	192, 194, 228, 416
COMMAND\setsidenotesep	61
COMMAND\setsidenotsep	363
COMMAND\setstanzaindent	302
COMMAND\setstanzaindents	51, 302, 360
COMMAND\setstanzapenalties	302
COMMAND\setstanzavalues	302
COMMAND\settoggle@series	247, 417, 421
COMMAND\showlemma	125, 414, 415
COMMAND\showwordrank	32, 135
COMMAND\sidenote@margin	415
COMMAND\sidenotemargin	60, 415, 420
COMMAND\sidenotesep	363
COMMAND\sidepstartnumtrue	19
COMMAND\skip	170
COMMAND\skipnumbering	23, 112, 123, 416, 423, 424
COMMAND\skipnumbering@reg	423
COMMAND\small	44
COMMAND\special	13
COMMAND\splitmaxdepth	166, 181
COMMAND\splitoff	179
COMMAND\splittopskip	165, 181, 182
COMMAND\stanza	22, 23, 52–54, 305, 363, 365, 426, 431
COMMAND\stanza@hang	304
COMMAND\stanza@line	304
COMMAND\stanzaindent	51, 302, 422

COMMAND\stanzaindent*	51
COMMAND\stanzaindentbase	301
COMMAND\stanzanumwrapper	54
COMMAND\startlock	22, 100, 123, 305
COMMAND\startstanzahook	363
COMMAND\startsub	22, 100, 121
COMMAND\stopmsd	310
COMMAND\stopmsdata	33, 308, 309
COMMAND\strip@pt	173
COMMAND\strutbox	181
COMMAND\sub@action	100, 110
COMMAND\sub@lock	98
COMMAND\sub@off	107, 264
COMMAND\sub@on	107, 264
COMMAND\subline@num	97, 98, 100
COMMAND\sublinenum@rep	414
COMMAND\sublinenumberstyle	23, 96, 414
COMMAND\sublinenumincrement	21
COMMAND\sublinenumr@p	96, 414, 416
COMMAND\sublinenumrep	96, 416
COMMAND\sublineref	55, 261, 267
COMMAND\subsectionmark	340
COMMAND\sw@inthisedtext	128
COMMAND\sw@list@inedtext	131, 138
COMMAND\symlinenum	364
COMMAND\symlinenum	363
COMMAND\sza@penalty	304
COMMAND>tag	422
COMMAND\text	358
COMMAND\text<language>	45
COMMAND\textcolor	72
COMMAND\textheight	71
COMMAND\the	414
COMMAND\thefootnoteA	34
COMMAND\thefootnoteX	418
COMMAND\thelabidx	296
COMMAND\thepage	105
COMMAND\thepstart	19
COMMAND\thepstartL	417
COMMAND\thepstartR	417
COMMAND\thestanza	53
COMMAND\this@line@list@version	117
COMMAND\thisfootnote	206
COMMAND\threecol@begin@insert	182
COMMAND\threecolfootfmt	182, 428
COMMAND\threecolfootfmtX	211
COMMAND\threecolfootgroup	180
COMMAND\threecolfootgroupX	211
COMMAND\threecolfootsetup	180
COMMAND\threecolfootsetupX	210

COMMAND\threecolvfootnote	181
COMMAND\threecolvfootnote@inserted	181
COMMAND\threecolvfootnoteX	211
COMMAND\toendnotes	26, 226, 429
COMMAND\twocolfootfmt	428
COMMAND\twocolfootfmtX	209
COMMAND\twocolfootgroupX	209
COMMAND\twocolfootsetupX	208
COMMAND\twocolvfootnoteX	209
COMMAND\twolines	246, 364
COMMAND\twolines@A	246
COMMAND\twolines@B	247
COMMAND\twolines@C	247
COMMAND\twolinesbutnotmore	364
COMMAND\twolinesonlyinsamepage	364
COMMAND\xtbeforeXnotes	364
COMMAND\uline	35
COMMAND\unhbox	174
COMMAND\unpenalty	175–177
COMMAND\unskip	176
COMMAND\unvxh	176, 365
COMMAND\unvxhX	365
COMMAND\upbracefill	333
COMMAND\usingcritext	359, 362
COMMAND\usingdtext	359, 362
COMMAND\vAfootnote	165
COMMAND\vadjust	120
COMMAND\variant	28
COMMAND\vbox	142, 144, 174, 179, 219
COMMAND\vfootnote	165, 170, 174, 181
COMMAND\vl@dbfnote	199, 415
COMMAND\vl@disnote	281
COMMAND\vl@dlsnote	281
COMMAND\vl@dosnote	281
COMMAND\vl@drsnote	281
COMMAND\vnumfootnoteX	416
COMMAND\vsizer	48, 71
COMMAND\vsplit	160
COMMAND\waklam	333
COMMAND\waklamec	333
COMMAND\wapunktel	333
COMMAND\wastricht	333
COMMAND\widthX	49, 366, 428
COMMAND\wrap@edcrossref	266, 421
COMMAND\wrapcontentX	45, 429
COMMAND\wrapped@bodyfootmarkX	217
COMMAND\wrapped@footfootmarkX	216
COMMAND\x...	56
COMMAND\xabslineref	268
COMMAND\xdef	90, 305



COMMAND\xflagref	56, 268, 366, 427
COMMAND\xleft@appenditem	90, 125
COMMAND\xlineref	56, 366, 427
COMMAND\xpageref	56
COMMAND\xpstartref	56, 419
COMMAND\xr	59
COMMAND\xright@appenditem	90
COMMAND\xsublineref	56
COMMAND\xxref	56, 269, 275, 419, 422, 429
COMMAND\zz@@@	414
ENVIRONMENTastanza	427
ENVIRONMENTedarrayc	338
ENVIRONMENTedarrayl	338
ENVIRONMENTedarrayr	338
ENVIRONMENTedtabularc	338
ENVIRONMENTedtabularl	338
ENVIRONMENTedtabularr	338
ENVIRONMENTledgroup	76, 287, 366, 427
ENVIRONMENTledgroupsize	288
PACKAGE(r)(e)ledmac	36
PACKAGEEledmac	12, 73, 100, 293, 361, 362, 422, 423
PACKAGEEledpar	423
PACKAGEEtoolbox	76
PACKAGEParallel	368
PACKAGEReledmac	365, 366
PACKAGEamsgen	316
PACKAGEamsmath	316, 317
PACKAGEbabel	45, 72, 327, 427
PACKAGEbiblatex	70
PACKAGEbidi	45, 46, 77, 426
PACKAGEcaption	84
PACKAGEcolor	71
PACKAGEcsquotes	431
PACKAGEedmac	1, 6, 10, 12–14, 73, 191, 198, 261, 302, 358, 368, 414
PACKAGEedstanza	1, 13, 300
PACKAGEeledmac	1, 10, 11, 14–16, 62, 198, 289, 293, 320, 342, 355, 359, 361–363, 418, 420, 422
PACKAGEeledpar	84, 166, 340, 368, 416, 420–422
PACKAGEetex	426
PACKAGEettoolbox	89, 90, 133, 233, 246, 255, 280, 341, 352
PACKAGEfancyhdr	258, 430
PACKAGEfloatrow	70, 366
PACKAGEfootmisc	34, 72, 77, 198, 368, 430
PACKAGEgeometry	15
PACKAGEglossaries	63, 298, 427
PACKAGEhandout	421
PACKAGEhyperlink	241
PACKAGEhyperref	56, 127, 216, 217, 262, 295, 344, 354, 419–421, 428, 431
PACKAGEifluatex	76
PACKAGEifxetex	76
PACKAGEimakeidx	61, 70, 77, 289, 293, 362, 418–420, 422

PACKAGEindextols	297
PACKAGEindextool	362
PACKAGEindextools	61, 70, 77, 85, 289, 293, 297, 362, 422, 427, 430
PACKAGEinputenc	135
PACKAGEledarab	72
PACKAGEledmac	1, 10, 13, 14, 72, 89, 293, 358, 359, 362, 365
PACKAGEledpar	72
PACKAGEMemoir	76, 293, 362, 368, 421
PACKAGEMorewrites	70
PACKAGEMusixtex	420
PACKAGEperpage	427
PACKAGEpolyglossia	43, 72, 164, 186, 427
PACKAGERagged2e	46, 76
PACKAGEReledmac	1, 2, 11, 12, 14–17, 20, 21, 24, 25, 28–30, 32, 34, 35, 38, 39, 42, 45, 46, 48, 50, 52, 54, 56, 57, 59, 61–64, 68, 70–74, 91, 93, 99, 100, 104, 107, 117, 126, 129, 130, 133, 158, 167, 170, 175, 186, 198, 223, 234, 238, 239, 247, 255, 266, 268, 270, 275, 276, 293, 313, 320, 340, 341, 354, 362, 363, 365–367, 425, 428, 430
PACKAGEReledpar	1, 4, 6, 8, 16, 19, 35, 49, 56, 58, 59, 68–70, 72, 74, 84, 91, 102, 107, 128, 130, 167, 171, 218, 219, 234, 240, 254, 256, 257, 289, 300, 426, 427, 429, 430
PACKAGESuffix	76
PACKAGEtabmac	1, 13, 368
PACKAGEulem	35
PACKAGEuninormalize	29
PACKAGExargs	29, 76
PACKAGExkeyval	74, 254
PACKAGEXI	5, 59, 276, 428
PACKAGEXref	275
PACKAGEXstring	76, 295

## A

\absline@num	1
Abu Kamil Shuja' b. Aslam	13
\actionlines@list	1
\actions@list	1
\add@inserts	1
\add@inserts@next	1
\add@msd@	1
\add@msdata	1
\add@msdata@firstlineofpage	1
\add@penalties	1
\add@Xgroupbyline	1
\addtol@denvbody	1
Adelard II	13
\advancelabel@refs	1
\advanceline	1, 22
\Aendnote	25
\affixline@num	1
\affixpstart@num	1
\affixside@note	1
\Afootnote	25

<code>\afternoteX</code>	46
<code>\afterruleX</code>	48
<code>\ampersand</code>	1, 54
<code>\append@notesX</code>	1
<code>\append@Xnotes</code>	1
<code>\applabel</code>	1, 57
<code>\appref</code>	1, 57
<code>\apprefwithpage</code>	1, 57
<code>\arrangementX</code>	1, 37
<code>\arrangementX@normal</code>	1
<code>\arrangementX@threecol</code>	1
<code>\arrangementX@twocol</code>	1
<code>\at@every@pend</code>	1
<code>\AtEndEveryPend</code>	1, 19
<code>\AtEveryPend</code>	1, 19
<code>\AtEveryPstart</code>	1, 19
<code>\AtEveryStanza</code>	1, 53
<code>\AtEveryStopStanza</code>	1, 53
<code>\AtStartEveryPstart</code>	1, 19
<code>\AtStartEveryStanza</code>	1, 53
<code>\autopar</code>	1, 18

**B**

<code>\ballast</code>	70
<code>\ballast@count</code>	1
Beeton, Barbara Ann Neuhaus Friend	18
<code>\beforeeledchapter</code>	1
<code>\BeforeEveryStopStanza</code>	1
<code>\beforeinsertingX</code>	45
<code>\beforenotesX</code>	48
<code>\beginnumbering</code>	1, 16
<code>\Bendnote</code>	25
<code>\Bfootnote</code>	25
<code>\bhookgroupX</code>	47
<code>\bhooknoteX</code>	45
<code>\bodyfootmarkA</code>	34
<code>\boxfootnotenumbers</code>	1
Bredon, Simon	13
Breger, Herbert	13, 320
Brey, Gerhard	13
Busard, Hubert L. L.	13
<code>\bypage@false</code>	1
<code>\bypage@true</code>	1
<code>\bypstart@false</code>	1
<code>\bypstart@true</code>	1

**C**

<code>\c@addcolcount</code>	1
<code>\c@ballast</code>	1
<code>\c@firstlinenum</code>	1

<code>\c@firstsublinenum</code>	1
<code>\c@labidx</code>	1
<code>\c@linenumincrement</code>	1
<code>\c@sublinenumincrement</code>	1
<code>\Cendnote</code>	25
<code>\Cfootnote</code>	25
<code>\ch@ck@l@ck</code>	1
<code>\ch@cksub@l@ck</code>	1
<code>\chapter</code>	1
<code>\check@pb@in@verse</code>	1
Chester, Robert of	13
Claassens, Geert H. M.	13
<code>\colalignX</code>	46
Copernicus, Nicolaus	13
<code>\critext</code>	358
<code>\ctab</code>	1
<code>\ctabtext</code>	1

## D

Dekker, Dirk-Jan	71
<code>\Dendnote</code>	25
<code>\Dfootnote</code>	25
<code>\disable@familiarnotes</code>	1
<code>\disable@notes</code>	1
<code>\disable@sidenotes</code>	1
<code>\disablel@dtabfeet</code>	1
<code>\do@actions</code>	1
<code>\do@actions@fixedcode</code>	1
<code>\do@actions@next</code>	1
<code>\do@ballast</code>	1
<code>\do@feet@custom@order</code>	1
<code>\do@feetX</code>	1
<code>\do@insidelinehook</code>	1
<code>\do@line</code>	1
<code>\do@linehook</code>	1
<code>\do@lockoff</code>	1
<code>\do@lockoffL</code>	1
<code>\do@lockon</code>	1
<code>\do@lockonL</code>	1
<code>\do@Xfeet</code>	1
<code>\doedindexlabel</code>	1
<code>\doendnotes</code>	1, 26
<code>\doendnotesbysection</code>	1, 26
<code>\doinsidelinehook</code>	1, 24
<code>\dolinehook</code>	1, 24
<code>\dosplits</code>	1
Downes, Michael	71, 174, 176
<code>\doextrafeet</code>	1
<code>\dummy@edindex</code>	1
<code>\dummy@edtext</code>	1

<code>\dummy@edtext@showlemma</code> .....	<u>1</u>
<code>\dummy@ref</code> .....	<u>1</u>

## E

<code>\edaftertab</code> .....	<u>1</u> , 66, 333
<code>edarrayc</code> (environment) .....	64
<code>edarrayl</code> (environment) .....	64
<code>edarrayr</code> (environment) .....	64
<code>\edatleft</code> .....	<u>1</u> , 66
<code>\edatright</code> .....	<u>1</u> , 66
<code>\edbeforetab</code> .....	<u>1</u> , 66, 333
<code>\edfilldimen</code> .....	<u>1</u>
<code>\edfont@info</code> .....	<u>1</u>
<code>\edindex</code> .....	<u>1</u> , 61
<code>\edindexlab</code> .....	<u>1</u> , 63
<code>\EDLABEL</code> .....	<u>1</u>
<code>\edlabel</code> .....	<u>1</u> , 55
<code>\edlabelE</code> .....	<u>1</u> , 57
<code>\edlabelS</code> .....	<u>1</u> , 57
<code>\edlabelSE</code> .....	<u>1</u> , 57
<code>\edlineref</code> .....	<u>1</u> , 55
<code>\edmakelabel</code> .....	<u>1</u> , 57
<code>\edpageref</code> .....	<u>1</u> , 55
<code>\edrowfill</code> .....	<u>1</u> , 65
<code>\EDTAB</code> .....	<u>1</u>
<code>\edtabcolsep</code> .....	<u>1</u> , 64
<code>\EDTABINDENT</code> .....	<u>1</u>
<code>\edtabindent</code> .....	<u>1</u>
<code>\EDTABtext</code> .....	<u>1</u>
<code>edtabularc</code> (environment) .....	64
<code>edtabularl</code> (environment) .....	64
<code>edtabularr</code> (environment) .....	64
<code>\EDTEXT</code> .....	<u>1</u>
<code>\edtext</code> .....	<u>1</u> , 24
<code>\edvertdots</code> .....	<u>1</u> , 66
<code>\edvertline</code> .....	<u>1</u> , 66
<code>\Eendnote</code> .....	25
<code>\Efootnote</code> .....	25
<code>\eled@chapter</code> .....	<u>1</u>
<code>\eled@section</code> .....	<u>1</u>
<code>\eled@sectioning@out</code> .....	<u>1</u>
<code>\eled@subsection</code> .....	<u>1</u>
<code>\eled@subsubsection</code> .....	<u>1</u>
<code>\eledchapter</code> .....	<u>1</u>
<code>\eledchapter*</code> .....	<u>1</u>
<code>\eledsection</code> .....	<u>1</u>
<code>\eledsection*</code> .....	<u>1</u>
<code>\eledsubsection</code> .....	<u>1</u>
<code>\eledsubsection*</code> .....	<u>1</u>
<code>\eledsubsubsection</code> .....	<u>1</u>

<code>\eledsubsubsection*</code>	1
<code>\enablel@dtabfeet</code>	1
<code>\end@lemmas</code>	1
<code>\endashchar</code>	1
<code>\endline@num</code>	1
<code>\endlock</code>	1, 22
<code>\endminipage</code>	1
<code>\endnumbering</code>	1, 16
<code>\endpage@num</code>	1
<code>\endprint</code>	1
<code>\endquotation</code>	1
<code>\endquote</code>	1
<code>\endsub</code>	1, 22
<code>\endsubline@num</code>	1
environments:	
<code>edarrayc</code>	64
<code>edarrayl</code>	64
<code>edarrayr</code>	64
<code>edtabularc</code>	64
<code>edtabularl</code>	64
<code>edtabularr</code>	64
<code>ledgroup</code>	54
<code>ledgroupsize</code>	54
<code>minipage</code>	54
<code>Euclid</code>	13
<code>\extensionchars</code>	1, 69

## F

<code>\f@x@l@cks</code>	1
Fairbairns, Robin	34
<code>\first@linenum@out@false</code>	1
<code>\first@linenum@out@true</code>	1
<code>\firstlinenum</code>	1, 21
<code>\firstseriesX@</code>	1
<code>\firstsublinenum</code>	1, 21
<code>\firstXseries@</code>	1
<code>\fix@page</code>	1
<code>\flag@end</code>	1
<code>\flag@end@later</code>	1
<code>\flag@end@RTL</code>	1
<code>\flag@start</code>	1
<code>\flag@start@later</code>	1
<code>\flag@start@RTL</code>	1
<code>\flagstanza</code>	1, 54
<code>\flush@notes</code>	1
<code>\fnpos</code>	1, 36
Folkerts, Menso	13
<code>\footfootmarkA</code>	34
<code>\footfudgefiddle</code>	1, 71
<code>\footnote</code>	1

\footnoteA	34
\footnoteB	34
\footnoteC	34
\footnoteD	34
\footnoteE	34
\footnotelang@lua	1
\footnotelang@poly	1
\footnoteoptions@	1
\footnoteXmark	35
\footnoteXtext	35
\footplitskips	1
\fullstop	1

## G

Gädeke, Nora	13
\get@edindex@hyperref	1
\get@edindex@ledinnote@command	1
\get@fnmark	1
\get@fnmarkX	1
\get@index@command	1
\get@linelistfile	1
\get@sw@txt	1
\get@thisfootnote	1
\get@thisfootnoteX	1
\getline@num	1
\gl@p	1

## H

\h@num	1
\hangindentX	45
\hidenumbering	1, 23
\hidenumberingonleftpage	1, 23
\hidenumberingonrightpage	1
\Hilfsbox	1
\hilfsbox	1
\hilfscount	1
\HILFSskip	1
\Hilfsskip	1
\hilfsskip	1
\hsizethreecolX	46
\hsizetwocolX	46
\Hy@raisedlink@left	1
\hyperlinkformat	1
\hyperlinkformatR	1
\hyperlinkR	1

## I

\if@addsw	1
\if@edtext@secondarg@	1
\if@eled@sectioning	1

<code>\if@firstlineofpage</code>	1
<code>\if@led@nofoot</code>	1
<code>\if@ledgroup</code>	1
<code>\if@lemmacommand@</code>	1
<code>\if@msdata@insertedfrompreviouspage</code>	1
<code>\if@noeled@sec</code>	1
<code>\if@noneed@Footnote</code>	1
<code>\if@RTL</code>	1
<code>\ifat@every@pend@star@</code>	1
<code>\ifat@every@pstart@star@</code>	1
<code>\ifautopar@pause</code>	1
<code>\ifbypage@</code>	1
<code>\ifbypage@R</code>	1
<code>\ifbypstart@</code>	1
<code>\ifbypstart@R</code>	1
<code>\ifeledmaccompat@</code>	1
<code>\iffirst@linenum@out@</code>	1
<code>\ifindtl@innote</code>	1
<code>\ifindtl@notenumber</code>	1
<code>\ifinserthangingsymbol</code>	1
<code>\ifinstanza</code>	1
<code>\ifl@d@dash</code>	1
<code>\ifl@d@elin</code>	1
<code>\ifl@d@esl</code>	1
<code>\ifl@d@pnum</code>	1
<code>\ifl@d@ssub</code>	1
<code>\ifl@d@Xmorethantwolines</code>	1
<code>\ifl@d@Xtwolines</code>	1
<code>\ifl@dend@X</code>	1
<code>\ifl@dhiddenumber</code>	1
<code>\ifl@dmemoir</code>	1
<code>\ifl@dpaging</code>	1
<code>\ifl@dpairing</code>	1
<code>\ifl@dprintingcolumns</code>	1
<code>\ifl@dprintingpages</code>	1
<code>\ifl@dskipnumber</code>	1
<code>\ifl@dskipversenumber</code>	1
<code>\ifl@dstartendok</code>	1
<code>\ifl@footmisc</code>	1
<code>\ifl@imakeidx</code>	1
<code>\ifl@indextools</code>	1
<code>\ifledfinal</code>	1, 69
<code>\ifledgroupnotesL@</code>	1
<code>\ifledgroupnotesR@</code>	1
<code>\iflednopbinverse</code>	1
<code>\ifledRcol</code>	1
<code>\ifledRcol@</code>	1
<code>\ifnocritical@</code>	1
<code>\ifnoend@</code>	1
<code>\ifnofamiliar@</code>	1



<code>\ifnoledgroup@</code> .....	1
<code>\ifnoquotation@</code> .....	1
<code>\ifnoteschanged@</code> .....	1
<code>\ifnumberedpar@</code> .....	1
<code>\ifnumbering</code> .....	1
<code>\ifnumberingR</code> .....	1
<code>\ifnumberline</code> .....	1
<code>\ifnumberstanza</code> .....	1
<code>\ifparapparatus@</code> .....	1
<code>\ifparledgroup</code> .....	1
<code>\ifpst@rtedL</code> .....	1
<code>\ifseriesbefore</code> .....	1
<code>\ifsidepstartnum</code> .....	1
<code>\ifstopmsdata@inserted@</code> .....	1
<code>\ifsublines@</code> .....	1
<code>\ifwidthliketwocolumns</code> .....	1
<code>\ifXendinsertsep@</code> .....	1
<code>\ifxindy@</code> .....	1
<code>\ifxindyhyperref@</code> .....	1
<code>\initnumbering@quote</code> .....	1
<code>\initnumbering@reg</code> .....	1
<code>\insert@count</code> .....	0, 1
<code>\insert@msdata</code> .....	1
<code>\insert@Xtxtbeforenotes</code> .....	1
<code>\inserthangingymbol</code> .....	1
<code>\insertlines@list</code> .....	1
<code>\insertparafootsepX</code> .....	1
<code>\inserts@list</code> .....	1

## J

Jayaditya .....	13
-----------------	----

## K

Kabelschacht, Alois .....	162
---------------------------	-----

## L

<code>\l@advance@parledgroup@beforenormalnotes</code> .....	1
<code>\l@d@add</code> .....	1
<code>\l@d@nums</code> .....	1
<code>\l@d@section</code> .....	1
<code>\l@d@set</code> .....	1
<code>\l@d@Xend</code> .....	1
<code>\l@dampcount</code> .....	1
<code>\l@dbfnote</code> .....	1
<code>\l@dcheckcols</code> .....	1
<code>\l@dcheckstartend</code> .....	1
<code>\l@dchset@num</code> .....	1
<code>\l@dcolcount</code> .....	1
<code>\l@dcollect@@body</code> .....	1
<code>\l@dcollect@body</code> .....	1

<code>\l@dcwidth</code>	1
<code>\l@dcnote</code>	1
<code>\l@dcnotetext</code>	1
<code>\l@dcnotetext@l</code>	1
<code>\l@dcnotetext@r</code>	1
<code>\l@ddodoreintrafeet</code>	1
<code>\l@dedbeginmini</code>	1
<code>\l@dedendmini</code>	1
<code>\l@emptyd@ta</code>	1
<code>\l@dend@close</code>	1
<code>\l@dend@open</code>	1
<code>\l@dend@stuff</code>	1
<code>\l@dend@Xfalse</code>	1
<code>\l@dend@Xtrue</code>	1
<code>\l@denbody</code>	1
<code>\l@desnote</code>	1
<code>\l@dfambeginmini</code>	1
<code>\l@dfamendmini</code>	1
<code>\l@dfeetbeginmini</code>	1
<code>\l@dfeetendmini</code>	1
<code>\l@dgetline@margin</code>	1
<code>\l@dgetlock@disp</code>	1
<code>\l@dgetref@num</code>	1
<code>\l@dgetsidenote@margin</code>	1
<code>\l@dobblearg</code>	1
<code>\l@disnote</code>	1
<code>\l@dlabel@parse</code>	1
<code>\l@dld@ta</code>	1
<code>\l@dlp@rbox</code>	1
<code>\l@dlsn@te</code>	1
<code>\l@dlsnote</code>	1
<code>\l@dmake@labels</code>	1
<code>\l@dmodforedtext</code>	1
<code>\l@dnullfills</code>	1
<code>\l@dnumpstartsL</code>	1
<code>\l@doldold@footnotetext</code>	1
<code>\l@dp@rsefootspec</code>	1
<code>\l@dparsedendline</code>	1
<code>\l@dparsedendpage</code>	1
<code>\l@dparsedendsub</code>	1
<code>\l@dparsedstartline</code>	1
<code>\l@dparsedstartpage</code>	1
<code>\l@dparsedstartsub</code>	1
<code>\l@dparsefootspec</code>	1
<code>\l@dpush@begins</code>	1
<code>\l@drd@ta</code>	1
<code>\l@dref@undefined</code>	1
<code>\l@drestorefills</code>	1
<code>\l@drestoreforedtext</code>	1
<code>\l@drp@rbox</code>	1

<code>\l@drsn@te</code> .....	1
<code>\l@drsnote</code> .....	1
<code>\l@dsetmaxcolwidth</code> .....	1
<code>\l@dskipnumberfalse</code> .....	1
<code>\l@dskipnumbertrue</code> .....	1
<code>\l@dtabaddcols</code> .....	1
<code>\l@dtabnoexpands</code> .....	1
<code>\l@dunboxmpfoot</code> .....	1
<code>\l@dunhbox@line</code> .....	1
<code>\l@dzeropenalties</code> .....	1
<code>\label</code> .....	57
<code>\labelpstartfalse</code> .....	1
<code>\labelpstarttrue</code> .....	1, 19
<code>\labelref@list</code> .....	1
<code>\labelrefsparseline</code> .....	1
<code>\labelrefsparsesubline</code> .....	1
<code>\last@page@num</code> .....	1
Lavagnino, John .....	12
<code>\led@check@nopb</code> .....	1
<code>\led@check@pb</code> .....	1
<code>\led@err@AutoparNotNumbered</code> .....	1
<code>\led@err@BadAction</code> .....	1
<code>\led@err@edtextoutsidepstart</code> .....	1
<code>\led@err@EdtextWithoutFootnote</code> .....	1
<code>\led@err@FootnoteNotInSecondArgEdtext</code> .....	1
<code>\led@err@HighEndColumn</code> .....	1
<code>\led@err@LineationInNumbered</code> .....	1
<code>\led@err@LowStartColumn</code> .....	1
<code>\led@err@ManyLeftnotes</code> .....	1
<code>\led@err@ManyRightnotes</code> .....	1
<code>\led@err@ManySidenotes</code> .....	1
<code>\led@err@NumberingNotStarted</code> .....	1
<code>\led@err@NumberingShouldHaveStarted</code> .....	1
<code>\led@err@NumberingStarted</code> .....	1
<code>\led@err@NumberingWithoutPstart</code> .....	1
<code>\led@err@PendNoPstart</code> .....	1
<code>\led@err@PendNotNumbered</code> .....	1
<code>\led@err@PstartInPstart</code> .....	1
<code>\led@err@PstartNotNumbered</code> .....	1
<code>\led@err@ReverseColumns</code> .....	1
<code>\led@err@toendnotes@outsidenumbering</code> .....	1
<code>\led@err@TooManyColumns</code> .....	1
<code>\led@err@UnequalColumns</code> .....	1
<code>\led@error@fail@patch@doclearpage</code> .....	1
<code>\led@error@fail@patch@iiiminipage</code> .....	1
<code>\led@error@fail@patch@makecol</code> .....	1
<code>\led@error@fail@patch@reinserts</code> .....	1
<code>\led@error@fail@patch@endminipage</code> .....	1
<code>\led@error@PackageAfterEledmac</code> .....	1
<code>\led@mess@NotesChanged</code> .....	1

<code>\led@mess@SectionContinued</code>	1
<code>\led@nopb</code>	1
<code>\led@nopbnum</code>	1
<code>\led@pb</code>	1
<code>\led@pb@setting</code>	1
<code>\led@pbnum</code>	1
<code>\led@reinit@index@fornote</code>	1
<code>\led@set@index@fornote</code>	1
<code>\led@toksa</code>	1
<code>\led@toksb</code>	1
<code>\led@warn@AppLabelOutSecondArgEdtext</code>	1
<code>\led@warn@BadAction</code>	1
<code>\led@warn@BadAdvancelineLine</code>	1
<code>\led@warn@BadAdvancelineSubline</code>	1
<code>\led@warn@BadLineation</code>	1
<code>\led@warn@BadLinenummargin</code>	1
<code>\led@warn@BadLockdisp</code>	1
<code>\led@warn@BadSetline</code>	1
<code>\led@warn@BadSetlinenum</code>	1
<code>\led@warn@BadSidenotemargin</code>	1
<code>\led@warn@BadSublockdisp</code>	1
<code>\led@warn@DuplicateLabel</code>	1
<code>\led@warn@edinde@outsidenumbering</code>	1
<code>\led@warn@LineFileObsolete</code>	1
<code>\led@warn@NoFile</code>	1
<code>\led@warn@NoIndexFile</code>	1
<code>\led@warn@NoMarginpars</code>	1
<code>\led@warn@RefUndefined</code>	1
<code>\led@warn@SeriesStillExist</code>	1
<code>\led@warning@hsizeX@deprecated</code>	1
<code>\led@warning@msdatawithoutstop</code>	1
<code>\led@warning@preXnotes@deprecated</code>	1
<code>\led@warning@Xhsize@deprecated</code>	1
<code>ledgroup (environment)</code>	54
<code>ledgroupsize (environment)</code>	54
<code>\ledinnernote</code>	1, 59
<code>\ledinnote</code>	1
<code>\ledinnotehyperpage</code>	1
<code>\ledinnotemark</code>	1
<code>\ledleftnote</code>	1, 60
<code>\ledlinenum</code>	1
<code>\ledllfill</code>	1
<code>\ledlsnotefontsetup</code>	1, 60
<code>\ledlsnotesep</code>	1, 60
<code>\ledlsnotewidth</code>	1, 60
<code>\lednopb</code>	1, 68
<code>\lednopbinversetrue</code>	69
<code>\lednopbnum</code>	1
<code>\ledouternote</code>	1, 59
<code>\ledpb</code>	1, 68

<code>\ledpbnum</code>	1
<code>\ledpbsetting</code>	1, 69
<code>\ledrightnote</code>	1, 60
<code>\ledrlfill</code>	1
<code>\ledrsnotefontsetup</code>	1, 60
<code>\ledrsnotesep</code>	1, 60
<code>\ledrsnotewidth</code>	1, 60
<code>\ledsectnomark</code>	1
<code>\ledsectnotoc</code>	1
<code>\ledsetnormalparstuff@common</code>	1
<code>\ledsetnormalparstuffX</code>	1
<code>\ledsidenote</code>	1, 60
<code>\leftctab</code>	1
<code>\leftlinenum</code>	1, 22
<code>\leftltab</code>	1
<code>\leftnoteupfalse</code>	60
<code>\leftpstartnum</code>	1
<code>\leftrtab</code>	1
Leibniz	13
<code>\lemma</code>	1, 27
<code>\letsforverteilen</code>	1
<code>\line@list</code>	1
<code>\line@list@stuff</code>	1
<code>\line@list@version</code>	1
<code>\line@margin</code>	1
<code>\line@num</code>	1
<code>\line@set</code>	1
<code>\lineation</code>	1, 21
<code>\linenum</code>	1, 27
<code>\linenum@out</code>	1
<code>\linenumberlist</code>	1, 21
<code>\linenumberstyle</code>	1, 23
<code>\linenumincrement</code>	1, 21
<code>\linenummargin</code>	1, 21
<code>\linenumr@p</code>	1
<code>\linenumrep</code>	1
<code>\linenumsep</code>	1, 22
<code>\linerrangesep@</code>	1
<code>\list@clear</code>	1
<code>\list@clearing@reg</code>	1
<code>\list@create</code>	1
<code>\lock@disp</code>	1
<code>\lock@off</code>	1
<code>\lock@on</code>	1
<code>\lockdisp</code>	1, 22
Lorch, Richard	13
<code>\ltab</code>	1
<code>\ltabtext</code>	1
Luecking, Dan	75

## M

<code>\m@mmf@check</code>	1
<code>\m@mmf@prepare</code>	1
<code>\M@ssect</code>	1
<code>\makehboxofhboxes</code>	1
<code>\managestanza@modulo</code>	1
<code>\maxhnotesX</code>	48
Mayer, Gyula	13
<code>\measurebody</code>	1
<code>\measuremcell</code>	1
<code>\measuremrow</code>	1
<code>\measuretbody</code>	1
<code>\measuretcell</code>	1
<code>\measuretrow</code>	1
Middleton, Thomas	13, 98
<code>minipage</code> (environment)	54
Mittelbach, Frank	12, 13
<code>\morenoexpands</code>	1, 71
<code>\mp@append@notesX</code>	1
<code>\mp@append@Xnotes</code>	1
<code>\mpfnpos</code>	1, 36
<code>\mpnormalfootgroup</code>	1
<code>\mpnormalfootgroupX</code>	1
<code>\mpnormalvfootnote</code>	1
<code>\mpnormalvfootnote@inserted</code>	1
<code>\mpnormalvfootnoteX</code>	1
<code>\mppara@footgroupX</code>	1
<code>\mppara@vfootnoteX</code>	1
<code>\mpparafootgroup</code>	1
<code>\mpparavfootnote</code>	1
<code>\mpthreecolfootgroup</code>	1
<code>\mpthreecolfootgroupX</code>	1
<code>\mpthreecolfootsetup</code>	1
<code>\mpthreecolfootsetupX</code>	1
<code>\mptwocolfootgroup</code>	1
<code>\mptwocolfootgroupX</code>	1
<code>\mptwocolfootsetup</code>	1
<code>\mptwocolfootsetupX</code>	1
<code>\msdata</code>	1, 33
<code>\msdata@c</code>	1
<code>\msdata@cR</code>	1
<code>\multfootsep</code>	1, 34
<code>\multiplefootnotemarker</code>	1

## N

<code>\n@num</code>	1
<code>\n@num@stanza</code>	1
<code>\new@line</code>	1
<code>\newhookarg@specific</code>	1
<code>\newhookcommand@series</code>	1

<code>\newhookcommand@series@reload</code> .....	1
<code>\newhooktoggle@series</code> .....	1
<code>\newhooktoggle@series@reload</code> .....	1
<code>\newhooktoggle@specific</code> .....	1
<code>\newseries@</code> .....	1
<code>\newverse</code> .....	1
<code>\NEXT</code> .....	1
<code>\no@expands</code> .....	1
<code>\noeledsec</code> .....	68
<code>\nomk@</code> .....	1
<code>\normal@footnotemarkX</code> .....	1
<code>\normal@page@break</code> .....	1
<code>\normal@pars</code> .....	1
<code>\normalbfnoteX</code> .....	1
<code>\normalbodyfootmarkX</code> .....	1
<code>\normalfootfmt</code> .....	1
<code>\normalfootfmtX</code> .....	1
<code>\normalfootfootmarkX</code> .....	1
<code>\normalfootgroup</code> .....	1
<code>\normalfootgroupX</code> .....	1
<code>\normalfootnoterule</code> .....	1
<code>\normalfootnoteruleX</code> .....	1
<code>\normalfootstart</code> .....	1
<code>\normalfootstartX</code> .....	1
<code>\normalvfootnote</code> .....	1
<code>\normalvfootnote@inserted</code> .....	1
<code>\normalvfootnoteX</code> .....	1
<code>\notefontsizeX</code> .....	44
<code>\notenumfontX</code> .....	44
<code>\noteschanged@false</code> .....	1
<code>\noteschanged@true</code> .....	1
<code>\noteswidthliketwocolumnsX</code> .....	49
<code>\nulledindex</code> .....	1
<code>\nullsetzen</code> .....	1
<code>\num@lines</code> .....	1
<code>\numberedpar@false</code> .....	1
<code>\numberedpar@true</code> .....	1
<code>\numberingfalse</code> .....	1
<code>\numberingtrue</code> .....	1
<code>\numberlinefalse</code> .....	20
<code>\numberlinetrue</code> .....	20
<code>\numberpstartfalse</code> .....	1, 19
<code>\numberpstarttrue</code> .....	1, 19
<code>\numberstanzafalse</code> .....	53
<code>\numberstanzatrue</code> .....	53
<code>\numlabfont</code> .....	1, 49

## O

<code>\old@hsize</code> .....	1
<code>\one@line</code> .....	1

optionauxdir	15, 429
optioncontinuousnumberingwithcolumns	428
optioninnnote	427
optioninnote	427
optionlinrangesep	254
optionnocritical	427
optionnoeledsec	346, 429
optionnoend	427
optionnopenalties	71
optionnotenumber	427

## P

\page@action	1
\page@num	1
\pagelinesep	1, 62
\pageref	57
\par@line	1
\para@footgroupX	1
\para@footsetup	1
\para@footsetupX	1
\para@vfootnoteX	1
\parafootfmt	1
\parafootfmtX	1
\parafootgroup	1
\parafootsepX	47
\parafootstart	1
\parafootstartX	1
\paravfootnote	1
\parindentX	45
\pausenumbering	1, 20
\pend	1, 17
Plato of Tivoli	13
\postbodyfootmark	1
\prebodyfootmark	1
\prenotesX	48
\prepare@edindex@fornote	1
\prepare@prenotesX	1
\prepare@Xgroupbyline	1
\prepare@Xprenotes	1
\prev@nopb	1
\prev@pb	1
\prevpage@num	1
\print@eledsection	1
\print@footnoteXrule	1
\print@leftmargin@eledsection	1
\print@lemma	1
\print@line	1
\print@notesX	1
\print@rightmargin@eledsection	1
\print@Xfootnoterule	1



<code>\print@Xnotes</code> .....	1
<code>\printendlines</code> .....	1
<code>\printlineendnote</code> .....	1
<code>\printlineendnotearea</code> .....	1
<code>\printlinefootnote</code> .....	1
<code>\printlinefootnotearea</code> .....	1
<code>\printlinefootnotenotenumbers</code> .....	1
<code>\printlines</code> .....	1
<code>\printnpnum</code> .....	1
<code>\printpstart</code> .....	1
<code>\printsymlineendnotearea</code> .....	1
<code>\printsymlinefootnotearea</code> .....	1
<code>\printXafternumber</code> .....	1
<code>\printXbeforenumber</code> .....	1
<code>\pstart</code> .....	1, 17
<code>\pstarteref</code> .....	1
<code>\pstartnum</code> .....	1
<code>\pstartref</code> .....	55

## Q

<code>\quotation</code> .....	1
<code>\quote</code> .....	1

## R

<code>\raggedX</code> .....	47
<code>\raw@text</code> .....	1
<code>\rbracket</code> .....	1
<code>\read@linelist</code> .....	1
<code>\ref</code> .....	57
<code>\ref@reg@later</code> .....	1
<code>\Relax</code> .....	1
<code>\reledmac@error</code> .....	1
<code>\reledmac@warning</code> .....	1
<code>\removehboxes</code> .....	1
<code>\reset@msd@options@iffullpage</code> .....	1
<code>\resetprevline@</code> .....	1, 101
<code>\resetprevpage@</code> .....	1
<code>\resetprevpage@num</code> .....	101
<code>\restore@familiarnotes</code> .....	1
<code>\restore@notes</code> .....	1
<code>\restore@sidenotes</code> .....	1
<code>\resumenumbering</code> .....	1, 20
<code>\rightctab</code> .....	1
<code>\rightlinenum</code> .....	1, 22
<code>\rightltab</code> .....	1
<code>\rightnoteupfalse</code> .....	60
<code>\rightrtab</code> .....	1
<code>\rightstartnum</code> .....	1
<code>\rigidbalance</code> .....	1
<code>\rigidbalanceX</code> .....	1

<code>\rtab</code> .....	1
<code>\rtabtext</code> .....	1

## S

Sacrobosco .....	13
<code>\sameword</code> .....	1, 28
<code>\sameword@inedtext</code> .....	1
Schöpf, Rainer .....	13
<code>\section@num</code> .....	1
<code>\select@@lemmafnt</code> .....	1
<code>\select@lemmafnt</code> .....	1, 49
<code>\Seref</code> .....	1, 57
<code>\Serefonlypage</code> .....	57
<code>\Serefwithpage</code> .....	1, 57
<code>\series</code> .....	1
<code>\seriesatbegin</code> .....	1, 35
<code>\seriesatend</code> .....	1, 35
<code>\set@line</code> .....	1
<code>\set@line@action</code> .....	1
<code>\set@Xtxtbeforenotes</code> .....	1
<code>\setapprefprefixmore</code> .....	58
<code>\setapprefprefixsingle</code> .....	58
<code>\setcommand@series</code> .....	1
<code>\sethangingsymbol</code> .....	1, 52
<code>\setistwofollowinglines</code> .....	1
<code>\setl@dlp@rbox</code> .....	1
<code>\setl@drpr@box</code> .....	1
<code>\setline</code> .....	1, 22
<code>\setlinenum</code> .....	1, 23
<code>\setmcellcenter</code> .....	1
<code>\setmcellleft</code> .....	1
<code>\setmcellright</code> .....	1
<code>\setmrowcenter</code> .....	1
<code>\setmrowleft</code> .....	1
<code>\setmrowright</code> .....	1
<code>\setmsdatalabel</code> .....	1, 34
<code>\setmsdataposition</code> .....	1, 34
<code>\setmsdataseries</code> .....	1, 34
<code>\setnoteswidthliketwocolumnsX@</code> .....	1
<code>\setnotesXpositionliketwocolumns@</code> .....	1
<code>\setprintendlines</code> .....	1
<code>\setprintlines</code> .....	1
<code>\setSErefonlypageprefixmore</code> .....	58
<code>\setSErefonlypageprefixsingle</code> .....	58
<code>\setSErefprefixmore</code> .....	58
<code>\setSErefprefixsingle</code> .....	58
<code>\setsidenotesep</code> .....	61
<code>\setstanzaindents</code> .....	1, 50
<code>\setstanzapenalties</code> .....	1, 51
<code>\setstanzavalues</code> .....	1

<code>\settccllcenter</code>	1
<code>\settccllleft</code>	1
<code>\settccllright</code>	1
<code>\settoggle@series</code>	1
<code>\setthrowcenter</code>	1
<code>\setthrowleft</code>	1
<code>\setthrowright</code>	1
<code>\setXnotespositionliketwocolumns@</code>	1
<code>\setXnoteswidthliketwocolumns@</code>	1
<code>\showlemma</code>	1, 69
<code>\showwordrank</code>	1, 32
<code>\sidenote@margin</code>	1
<code>\sidenotemargin</code>	1, 60
<code>\sidepstartnumtrue</code>	19
<code>\skip@lockoff</code>	1
<code>\skipnumbering</code>	1, 23
<code>\splitoff</code>	1
<code>\spreadmath</code>	1, 64
<code>\spreadtext</code>	1, 64
<code>\stanza</code>	1, 50
<code>\stanza@count</code>	1
<code>\stanza@hang</code>	1
<code>\stanza@line</code>	1
<code>\stanzaindent</code>	1, 51
<code>\stanzaindent*</code>	1, 51
<code>\stanzaindentbase</code>	1, 50
<code>\stanzanumwrapper</code>	1, 54
<code>\startlock</code>	1, 22
<code>\startsub</code>	1, 22
<code>\stepl@dcclcount</code>	1
<code>\stopmsdata</code>	1, 33
<code>\strip@szacnt</code>	1
<code>\sub@action</code>	1
<code>\sub@lock</code>	1
<code>\sub@off</code>	1
<code>\sub@on</code>	1
<code>\subline@num</code>	1
<code>\sublinenumberstyle</code>	1, 23
<code>\sublinenumincrement</code>	1, 21
<code>\sublinenumr@p</code>	1
<code>\sublinenumrep</code>	1
<code>\sublineref</code>	1, 55
<code>\sublines@false</code>	1
<code>\sublines@true</code>	1
<code>\sublock@disp</code>	1
<code>\sublockdisp</code>	1
Sullivan, Wayne	13, 70, 84, 88, 174, 175, 261, 300
<code>\sza@penalty</code>	1

## T

<code>\tabHilfbox</code>	1
<code>\tabhilfbox</code>	1
<code>\theadcolcount</code>	1
<code>\theadtext</code>	1
<code>\theendpageline</code>	1
<code>\thefootnoteA</code>	34
Theodosius	13
<code>\thepageline</code>	1
<code>\thepstart</code>	1, 19
<code>\thestanza</code>	1, 53
<code>\thestartpageline</code>	1
<code>\this@line@list@version</code>	1
<code>\threecol@begin@insert</code>	1
<code>\threecolfootfmt</code>	1
<code>\threecolfootfmtX</code>	1
<code>\threecolfootgroup</code>	1
<code>\threecolfootgroupX</code>	1
<code>\threecolfootsetup</code>	1
<code>\threecolfootsetupX</code>	1
<code>\threecolvfootnote</code>	1
<code>\threecolvfootnote@inserted</code>	1
<code>\threecolvfootnoteX</code>	1
<code>\toendnotes</code>	1, 26
<code>\toendnotes*</code>	1
<code>\twocolfootfmt</code>	1
<code>\twocolfootfmtX</code>	1
<code>\twocolfootgroup</code>	1
<code>\twocolfootgroupX</code>	1
<code>\twocolfootsetup</code>	1
<code>\twocolfootsetupX</code>	1
<code>\twocolvfootnote</code>	1
<code>\twocolvfootnote@inserted</code>	1
<code>\twocolvfootnoteX</code>	1

## U

<code>\unvxhX</code>	1
----------------------	---

## V

Vamana	13
<code>\variab</code>	1
<code>\vbfnoteX</code>	1
<code>\vl@dbfnote</code>	1
<code>\vl@dcsnote</code>	1
<code>\vl@disnote</code>	1
<code>\vl@dlsnote</code>	1
<code>\vl@dosnote</code>	1
<code>\vl@drsnote</code>	1
<code>\vnumfootnoteX</code>	1

## W

Whitney, Ron .....	12
\widthX .....	49
\wrap@edcrossref .....	1
\wrapcontentX .....	45
\wrapped@bodyfootmarkX .....	1
\wrapped@footfootmarkX .....	1
Wujastyk, Dominik .....	12

## X

\X@atbegininsertion .....	1
\X@beforeinsertion .....	1
\X@doreinfeet .....	1
\xabslineref .....	1
\Xafterlemmaseparator .....	43
\Xafternote .....	46
\Xafternumber .....	41
\Xafterrule .....	48
\Xaftersymlinenum .....	41
\Xarrangement .....	1, 37
\Xarrangement@normal .....	1
\Xarrangement@paragraph .....	1
\Xarrangement@threecol .....	1
\Xarrangement@twocol .....	1
\Xbeforeinserting .....	45
\Xbeforelemmaseparator .....	43
\Xbeforenotes .....	47
\Xbeforenumber .....	39, 41
\Xbeforesymlinenum .....	41
\Xbhookgroup .....	47
\Xbhooknote .....	45
\Xboxlinenum .....	41
\Xboxlinenumalign .....	42
\Xboxsymlinenum .....	41
\Xcolalign .....	46
\xedindex .....	1
\xedlabel .....	1
\xedtext .....	1
\Xendafterenumber .....	41
\Xendafterlemmaseparator .....	43
\Xendafternote .....	49
\Xendafterpagenumber .....	42
\Xendaftersymlinenum .....	41
\Xendahookinplaceofnumber .....	43
\Xendahooklinenumber .....	43
\Xendbeforelemmaseparator .....	43
\Xendbeforenumber .....	41
\Xendbeforepagenumber .....	42
\Xendbeforesymlinenum .....	41
\Xendbhookinplaceofnumber .....	43

\Xendbhooklinenumber .....	42
\Xendbhooknote .....	45
\Xendboxendlinenumalign .....	42
\Xendboxlinenum .....	42
\Xendboxlinenumalign .....	42
\Xendboxstartlinenumalign .....	42
\Xendboxsymlinenum .....	42
\Xendhangindent .....	45
\Xendinplaceoflemmaseparator .....	43
\Xendinplaceofnumber .....	41
\Xendinplaceofpagenumber .....	39
\Xendlemmadisablefontselection .....	44
\Xendlemmafont .....	44
\Xendlemmaseparator .....	43
\Xendlineprefixmore .....	42
\Xendlineprefixsingle .....	42
\Xendlinerangeseparator .....	39
\Xendmorethantwolines .....	40
\Xendnonumber .....	40
\Xendnotefontsize .....	44
\Xendnotenumfont .....	44
\Xendnumberonlyfirstinline .....	38
\Xendnumberonlyfirstintwolines .....	38
\Xendpagenumberonlyfirst .....	38
\Xendparagraph .....	49
\Xendsep .....	49
\Xendsublinesep .....	40
\Xendsymlinenum .....	38
\Xendsympagenum .....	39
\Xendtwolines .....	40
\Xendtwolinesbutnotmore .....	40
\Xendwrapcontent .....	45
\xflagref .....	1
\Xgroupbyline .....	47
\Xgroupbylineseparetwolines .....	47
\Xhangindent .....	45
\Xhsizethreecol .....	46
\Xhsizetwocol .....	46
\Xinplaceoflemmaseparator .....	43
\Xinplaceofnumber .....	41
\Xinsertparafootsep .....	1
\Xledsetnormalparstuff .....	1
\xleft@appenditem .....	1
\Xlemmadisablefontselection .....	44
\Xlemmafont .....	44
\Xlemmaseparator .....	43
\Xlinerrangeseparator .....	39
\xlineref .....	1, 55
\Xmaxhnotes .....	48
\Xmorethantwolines .....	39

\Xnolemmaseparator	1, 43
\Xnonbreakableafternumber	41
\Xnonumber	40
\Xnotefontsize	44
\Xnotenumfont	44
\Xnoteswidthliketwocolumns	49
\Xnumberonlyfirstinline	38
\Xnumberonlyfirstintwolines	38
\Xonlypstart	40
\Xpagelinesep	41
\xpageref	1, 55
\Xparafootsep	47
\Xparindent	45
\Xprenotes	1, 48
\Xprenotes@	1
\Xpstart	40
\Xpstarteverytime	40
\xpstartref	1, 55
\XR@test	1
\XR@test@mac	1
\XR@test@mac@test	1
\Xragged	47
\xright@appenditem	1
\Xrigidbalance	1
\Xstanza	40
\Xstanzaseparator	40
\Xstorelineinfo	1
\xsublineref	1, 55
\Xsublinesep	40
\Xsymlinenum	38
\Xtoendnotes	26
\Xtwolines	39
\Xtwolinesonlyinsamepage	39
\Xtxtbeforenotes	47
\Xunvxh	1
\Xwidth	49
\Xwrapcontent	45
\Xwrapendlemma	45
\Xwraplemma	44
\xxref	1, 56

## Z

\zz@@@	1
--------	---

## Change History

v0.1.0.	
General: First public release	1
v0.2.0.	
\ifl@dmemoir: Added \ifl@dmemoir for memoir class having been used	76
\morenoexpands: Added \l@dtabnoexpands to \no@expands	126
\reledmac@error: Added \eledmac@error and replaced error messages	78
General: Added tabmac code, and extended indexing	1
v0.2.1.	
\@lab: Removed page setting from \@lab	264
\doxtrafeet: Renamed \doxtrafeet to \l@ddoxtrafeet	255
\edlabel: Tweaked \edlabel to get correct page numbers	261
\l@ddodoreinxtrafeet: Renamed \dodoreinxtrafeet to \l@ddodoreinxtrafeet	256
\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	126
\zz@@@: Minor change to \zz@@@	261
General: Added text about normal labeling	57
Bug fixes and match with mempatch v1.8	1
Major changes to insert code when memoir is loaded	258
v0.2.2.	
\footfudgefiddle: Added \footfudgefiddle	173
\line@list@stuff: Added initial write of page number in \line@list@stuff	118
\para@footsetup: Added \footfudgefiddle to \para@footsetup	173
\para@footsetupX: Added \footfudgefiddle to \para@footsetupX	213
General: Improved paragraph footnotes	1
New Dekker example	1
Used \providecommand for \@gobblethree to avoid clash with the amsfonts package	84
v0.3.0.	
\@lab: Replaced \the\line@num by \linenumr@p\line@num in \@lab, and similar for sub-lines	264
\@nl@reg: Added a bunch of code to \@nl for handling \setlinenum	105
\ledlinenum: Added \linenumr@p and \sublinenum@rep to \leftlinenum and \rightlinenum	96
\linenumberlist: Added \linenumberlist mechanism	84
\printendlines: Added \linenumr@p and \sublinenumr@p to \printendlines	230
\printlines: Added \linenumr@p and \sublinenumr@p to \printlines	196
\sublinenumr@p: Added \linenumberstyle and \sublinenumberstyle	96
General: Includes edstanza and more	1
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 critical footnotes	286
\Xarrangement@normal: Added minpage footnote setup to \footnormal	167
\edtext: Added \showlemma to \edtext	127
\l@dfeetendmini: Added \l@dfeetbeginmini, \l@dfeetendmini and all their supporting code	284



\mpnormalfootgroup: Added \mpnormalfootgroup	171
\mpnormalvfootnote: Added \mpnormalvfootnote	169
\showlemma: Added \showlemma	84
General: Added final/draft options	74
Added ledgroup environment	287
Added ledgroupsize environment	288
Added minipage, etc., support	1
v0.4.1.	
\do@Xfeet: Changed \do@Xfeet code for easier extensions	256
\edindex: Let eledmac take advantage of memoir's indexing	293
\print@Xnotes: Added \op@Xfeet	256
General: Added code for changing \doclearpage	259
Not released. Minor editorial improvements and code tweaks	1
Only change \@footnotetext and \@footnotemark if memoir not used	198
v0.5.0.	
\@footnotetext: Enabled regular \footnote in numbered text	199
\@xympar: Eliminated \marginpar disturbance	276
General: Added left and right side notes	277
Added sidenotes, familiar footnotes in numbered text	1
v0.5.1.	
\affixline@num: Changed \affixline@num to cater for sidenotes	153
\l@edgetsidenote@margin: Added \sidenotemargin and \sidenote@margin	277
General: Added moveable side note	277
Fixed right line numbers killed in v0.5	1
Only change \hsize in ledgroupsize environment otherwise page number can be in wrong place	288
v0.6.0.	
\@lopR: Added \@pend, \@pendR, \@lopL and \@lopR in anticipation of parallel processing	107
\@nl@reg: Added \fix@page to \@nl	105
Extended \@nl to include the page number	105
\fix@page: Added \last@page@num and \fix@page	106
\get@thisfootnote: Changed \l@dbfnote and \vl@dbfnote as originals could give incorrect markers in the footnotes	199
\new@line: Extended \new@line to output page numbers	118
General: Fixed long paragraphs looping	1
Fixed minor typos	1
Prepared for eledpar package	1
v0.7.0.	
\@nl@reg: Added \@nl@reg	105
\@ref@reg: Added \@ref@reg	114
\affixline@num: Added skipnumering to \affixline@num	153
\do@actions@fixedcode: Added \do@actions@fixedcode	152
\do@actions@next: Added number skipping to \do@actions	151
\do@insidelinehook: Added \do@linehook for use in \do@line	149
\endnumbering: Changed \endnumbering for eledpar	88
\fix@l@cks: Added \ch@cksub@l@ck, \ch@ck@l@ck and \fix@l@cks	156
\footplitskips: Added \footplitskips for use in many footnote styles	166
\get@linelistfile: Added \get@linelistfile	104
\initnumbering@reg: Added \initnumbering@reg	86

\l@advance@parledgroup@beforenormalnotes: Added \l@dunboxmpfoot containing some common code	287
\l@dcsnotetext@r: Added \l@emptyd@ta	149
\l@dgetline@margin: Added \l@dgetline@margin	92
\l@dgetlock@disp: Added \l@dgetlock@disp	95
\l@dgetsidenote@margin: Added \l@dgetsidenote@margin	277
\l@dnumpstartsL: Added \l@dnumpstartsL, \ifl@dpairing and \ifpst@rted for/from eledpar	84
\l@drsn@te: Added \l@dlsn@te and \l@drsn@te for use in \do@line	149
\l@dzeropenalties: Added \l@dzeropenalties	144
\ledlinenum: Added \ledlinenum for use by \leftlinenum and \rightlinenum	96
\line@list@stuff: Deleted \page@start from \line@list@stuff	118
\list@clearing@reg: Added \list@clearing@reg	103
\n@num: Added \n@num	112
\normalbfnoteX: Removed extraneous space from \normalbfnoteX	205
\resumenumbering: Changed \resumenumbering for eledpar	89
\setprintendlines: Added \setprintendlines for use by \printendlines	228
\setprintlines: Added \setprintlines for use by \printlines	192
\skipnumbering: Added \skipnumbering and supports	123
\sublinenumincrement: Added \firstlinenum, \linenumincrement, \firstsublinenum and \linenumincrement	94
\sublinenumr@p: Using \linenumrep instead of \linenumr@p	96
Using \sublinenumrep instead of \sublinenumr@p	96
\vnumfootnoteX: Removed extraneous space from \vnumfootnoteX	207
General: eledmac having been available for 2 years, deleted the commented out original edmac texts	1
Maieul Rouquette new maintainer	1
Made macros of all messages	77
Replaced all \interAfootnotelinepenalty, etc., by just \interfootnotelinepenalty	1
Tidying up for eledpar and ledarab packages	1
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.	
\l@dnumpstartsL: Added \ifledRcol and \ifnumberingR for/from eledpar	84
General: For compatibility with eledpar, possibility to use \autopar on the right side.	1
Possibility to number \pstart.	19

Possibility to number the pstart with the commands <code>\numberpstarttrue</code> . . . . .	1
v0.12.1.	
General: Don't number <code>\pstarts</code> of stanza. . . . .	1
The numbering of <code>\pstarts</code> restarts on each <code>\beginnumbering</code> . . . . .	1
v0.13.0.	
<code>\managestanza@modulo</code> : New <code>stanzaindentsrepetition</code> counter to repeat stanza indents every $n$ verses. . . . .	302
General: New <code>stanzaindentsrepetition</code> counter to repeat stanza indents every $n$ verses. . .	51
New <code>stanzaindentsrepetition</code> counter: to repeat stanza indents every $n$ verses. . . . .	1
v0.13.1.	
General: <code>\thepstartL</code> and <code>\thepstartR</code> use now <code>\bfseries</code> and not <code>\bf</code> , which is deprecated and makes conflicts with memoir class. . . . .	1
v0.14.0.	
<code>\edlabel</code> : Tweaked <code>\edlabel</code> to get correct line number if the command is first element of a paragraph. . . . .	261
General: Tweaked <code>\edlabel</code> to get correct line number if the command is first element of a paragraph. . . . .	1
v0.15.0.	
<code>\affixline@num</code> : Line numbering can be disabled. . . . .	153
<code>\ifinserthangingsymbol</code> : New management of <code>hangingsymbol</code> insertion, preventing undesirable insertions. . . . .	300
<code>\printlines</code> : Line numbering can be reset at each <code>pstart</code> . . . . .	195
General: Line numbering can be reset at each <code>pstart</code> . . . . .	91
Possibility to print <code>\pstart</code> number inside. . . . .	19
v0.17.0.	
<code>\ifinserthangingsymbol</code> : New new management of <code>hangingsymbol</code> insertion, preventing undesirable insertions. . . . .	300
v1.0.0.	
<code>\morenoexpands</code> : Change to be compatible with new features . . . . .	126
General: <code>\lemma</code> can contain commands. . . . .	27
Debug in lineation command . . . . .	21
New generic commands to customize footnote display. . . . .	36, 246
Options <code>nonum</code> and <code>nosep</code> in <code>\Xfootnote</code> . . . . .	25
Options of <code>\Xfootnotes</code> . . . . .	163
Possibility to have commands in sidenotes. . . . .	60
Some compatibility break with <code>eledmac</code> . Change of name: <code>eledmac</code> . . . . .	1
v1.0.1.	
General: Correction on <code>\Xnumberonlyfirstinline</code> with lineation by <code>pstart</code> or by page. . .	38
v1.1.0.	
<code>\Xprenotes</code> : New skip <code>\Xprenotes@</code> . . . . .	221
<code>\settoggle@series</code> : <code>\settoggle@series</code> switch the global value of the toggle, not only the local value. . . . .	247
General: Add <code>\labelpstarttrue</code> . . . . .	19
Add <code>\Xnumberonlyfirstintwolines</code> . . . . .	38
Add <code>\Xpstart</code> and <code>\Xonlypstart</code> . . . . .	40
New hook to add arbitrary code at the beginning of the notes . . . . .	45
New options for block of notes. . . . .	47
New package option: <code>parapparatus</code> . . . . .	1
New tools to change order of series . . . . .	245
Sectioning commands. . . . .	67

v1.2.0.	
\Xprenotes:	Debug in familiar footnotes (bug introduced by v1.1). . . . . 221
\endquote:	Compatibility of \ledchapter with the <i>memoir</i> class. . . . . 339
v1.3.0.	
\endquote:	<i>Quotation</i> and quote environment inside numbered sections. . . . . 339
v1.4.0.	
\edtext:	Compatibility of \edtext with the right-to-left direction (with Polyglossia). 127
\ledsetnormalparstuffX:	Direction of footnotes with polyglossia. . . . . 217
\newseries@:	Remembers the language of the lemma, in order to create a correct direction for the footnote separator. . . . . 236
\rbraacket:	Switch the right bracket to a left bracket when the lemma is RTL (needs polyglossia or LuaTeX). . . . . 186
General:	Compatibility with LuaTeX of RTL notes. . . . . 1
v1.4.1.	
\affixside@note:	Remove spurious spaces. . . . . 283
\endquote:	New option <i>noquotation</i> . . . . . 339
\get@thisfootnote:	Compatibility of standard footnotes with eledmac when these footnotes contain any commands. . . . . 199
\labelrefsparsesubline:	Fix bug with \edlabel. . . . . 263
v1.4.2.	
General:	Debug with some special classes. . . . . 1
v1.4.3.	
General:	Add \Xnonbreakableafternumber. . . . . 41
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. . . . . 339
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. . . . . 205
v1.5.0.	
\do@insidelinehook:	Added \do@insidelinehook for use in \do@oline . . . . . 149
\edindex:	Compatibility with imakeidx package, and possibility to use multiple index with \edindex. . . . . 293
General:	Correct indexing when the call is made in critical notes. . . . . 289
v1.5.1.	
\managestanza@modulo:	Correct stanzaindentsrepetition counter . . . . . 302
\normalvfootnoteX:	Fix bug with normal familiar footnotes when mixing RTL and LTR text. . . . . 202
v1.6.0.	
\newverse:	Add \falseverse macro. . . . . 305
v1.6.1.	
\AtStartEveryPstart:	Spurious space in \pstart. . . . . 140

\ifinserthangingsymbol: Hang verse is now not automatically flush right. . . . .	300
\l@dunhbox@line: Move the call to \inserthangingsymbol to allow use \hfill inside. . . . .	145
\pend: Spurious space in \pend. . . . .	142
General: Corrects a false hanging verse when a verse is exactly the length of a line. . . . .	1
v1.7.0.	
General: New features for managing page breaks. . . . .	68
v1.8.0.	
\endquote: Correction of sectioning commands in parallel texts. . . . .	339
\get@index@command: Debug \get@index@command and compatibility with hyperref package. . . . .	292
\newhookcommand@series@reload: Debug \beforenotesX and \maxhnotesX which did not work. . . . .	249
\prevpage@num: Correct \parafootsep when using with ledgroup. . . . .	178
General: Compatibility with parledgroup option ofeledpar package. . . . .	1
If imakeidx and hyperref are loaded, adds hyperref in the index. . . . .	289
v1.8.1.	
General: Debug endnotes when more than one series is used (change the position where tools for endnotes are defined). . . . .	222
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, witheledpar. . . . .	289
v1.9.0.	
\doxtrafeet: Add \fnpos to choice the order of footnotes. . . . .	255
\l@dfeetendmini: Add \mpfnpos to choice the order of footnotes in minipage / ledgroup. . . . .	284
v1.10.0.	
\endquote: Correction of sectioning commands in parallel texts. . . . .	339
General: Add \pstartref and \xpstartref to refer to a pstart number (extension of \edlabel). . . . .	1
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. . . . .	44
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. . . . .	105
\AtStartEveryPstart: New optional argument for \pstart, to execute code before it. . . . .	140
\edindex: Use correctly default index when imakeidx is loaded. . . . .	293
\endquote: \ledxxx sectioning commands are deprecated and replaced by \eledxxx commands. . . . .	339
\initnumbering@reg: \beginnumbering is defined only on eledmac, not on eledpar. . . . .	86
\l@dgetsidenote@margin: \sidenotemargin is now directly defined in eledmac to be able to manage eledpar. . . . .	277
\l@disnote: \l@dlsnote, \l@drsnote and \l@dcsnote defined only one time, in eledmac, including needs for eledpar case. . . . .	278
\l@dnumpstartsL: Add \ifledRcol@ for eledpar . . . . .	84
\l@dunhbox@line: \do@line is split in more little commands. . . . .	146
\newhookcommand@series@reload: Debug \beforenotesX and \maxhnotesX which did not work when called after \footparagraphX. . . . .	249
Debug \Xbeforenotes and \Xmaxhnotes which did not work when called after \footparagraph. . . . .	249
\pend: New optional argument for \pend, to execute code after it. . . . .	142
\stanza: &can have an optional argument: content to be printed after. . . . .	305
\Stanza can have an optional argument: content to be printed before. . . . .	305
Add \newverse macro, \falseverse deprecated. . . . .	305
General: Add \ledinnernote and \ledouternote commands. . . . .	60
Add \Xendparagraph and related settings. . . . .	49
Add hyperlink to crossref (needs hyperref package). . . . .	55
Compatibility with musixtex. . . . .	1
Debug eledmac sectioning command after using \resumenumbering. . . . .	1
Ensure that imakeidx is loaded <i>before</i> eledmac . . . . .	289
New hooks: \Xafterrule and \afterruleX . . . . .	48
New options for ragged-paragraph notes . . . . .	47
New sectioning commands. . . . .	67
Optional arguments for \pstart and \pend. . . . .	18

## v1.12.1.

\wrap@edcrossref: Fix spurious spaces. . . . .	266
--	-----

## v1.12.2.

\l@dunhbox@line: Fix a bug with critical notes at the tops of pages (added by v12.0.0) . . . . .	145
--	-----

## v1.12.3.

\flag@end: \flag@start and \flag@end are now defined only one time for eledmac and eledpar . . . . .	119
\flag@start send a error message when a \edtext is done without insert (note) . . . . .	119
\reledmac@error: Replaced error messages . . . . .	78
General: Add macros for new messages since v0.7 . . . . .	77
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. . . . .	60
Debug left and right notes (bugs added by 1.12.0) . . . . .	1
Underline lemma in \eledxxx when using draft mode. . . . .	1

## 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	295
\@ssect: Debug \eledchapter in parallel with memoir	342
\doinsidelinehook: Added \dolinehook and \doinsidelinehook	149
\endnumbering: Allow to mix parallel columns and normal text when using	
\pausenumbering	88
\l@dgobblearg: \l@dgobblearg becomes \l@gobbelloptarg	322
\l@drestoreforedtext: Debug optional arguments of \Xfootnote in tabular context	322
\resumenumbering: Debug \resumenumbering	89
v1.12.7.	
\wrap@edcrossref: \wrap@edcrossref is now robust	266
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	119
v1.13.0.	
\newhooktoggle@series@reload: Add \newhookcommand@toggle@reload	249
\para@footsetupX: In \para@footsetupX, use \columnwidth instead of \hsize	213
\settoggle@series: \settoggle@series can take an optional arguments to reload	
series setup.	247
General: Add \Xnoteswidthliketwocolumns and \noteswidthliketwocolumnsX	49
Added widthliketwocolumns option	74
v1.13.1.	
\ifat@every@pstart@star@: Add \l@dzeroopenalties in \pstart	140
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
v1.13.2.	
\l@dnumpstartsL: Add \ifl@dpageing for \eledpar	84
General: Fix bug with normal footnotes, added by v1.13.0.	1
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.	
\edindex: Let \eledmac take advantage of \makeidx even when memoir class is used	293
General: Debug spurious characters before endnotes.	222
Delete previous override of \l@d@wrindexhyp at the beginning of a document	
when hyperref is not loaded.	297
Move gobbling command	84
Provide \@gobblefour	84
v1.14.1.	
\@ssect: Debug sectioning commands when using both <b>handout</b> and <b>hyperref</b>	
package.	344
v1.14.2.	
\@ssect: Debug \edtext after starred sectioning commands when using memoir class.	342
v1.15.0.	
\@edtext@level: New boolean \if@edtext@.	127
\arrangementX@threecol: Correct bug with paragraphed familiar footnotes setting.	212
\endsub: Restore subline feature (disabled by mistake in v1.8.0).	121
\if@lemmacommand@: New boolean \iflemmacommand@.	132

General: Fix bug with footnotes layout when using some options of the geometry package (bug add by v1.13.0). . . . .	1
New commands <code>\AtEveryPstart</code> and <code>\AtEveryPend</code> . . . . .	19
New tools to prevent ambiguous references in lemma . . . . .	28
v1.15.1.	
<code>\line@list@stuff</code> : Revert modification of 1.5.2 which makes bug with numbering. Leave vertical mode to solve spurious space before minipage. . . . .	118
v1.16.0.	
General: <code>\edtext</code> is now defined only in <code>eledmac</code> , not in <code>eledpar</code> . Debug wrong numbering when using <code>\sameword + eledpar + \tag</code> command. . . . .	127
Compatibility of standard footnotes with some biblatex styles. . . . .	1
New <code>\stanzaindent</code> command. . . . .	1
v1.16.1.	
<code>\xlineref</code> : <code>\lineref</code> is not defined if defined by some other package, like <code>lineno</code> . <code>Eledmac</code> provides <code>\edlineref</code> instead. . . . .	267
v1.17.0.	
<code>\edtext</code> : Error message when calling <code>\edtext</code> outside of a numbered paragraph. . . .	127
v1.18.0.	
<code>\@edindex@hyperref</code> : Fix spurious space with <code>\edindex</code> when using <code>imakeidx/indextools + hyperref</code> . . . . .	295
<code>\edlabel</code> : <code>\edlabel</code> is now defined only one time for both <code>eledmac</code> and <code>eledpar</code> . . .	261
<code>\l@d@section</code> : Option <code>parapparatus</code> works for endnotes. . . . .	223
<code>\l@dnumpstartsL</code> : Add <code>\ifl@dprintingpages</code> and <code>\@dprintingcolumns</code> for <code>eledpar</code> . . . . .	84
<code>\print@line</code> : Compatibility with <code>Lua<math>\TeX</math></code> RTL languages. . . . .	146
<code>\printlinefootnote</code> : Code refactoring in <code>\printlinefootnote</code> : the printing of the numbers are factorized in <code>\printlinefootnotearea</code> . . . . .	188
<code>\printpstart</code> : Debug <code>\Xpstart</code> with parallel pages and columns ( <code>eledpar</code> ) . . . . .	187
General: Add <code>\Xpstarteverytime</code> . . . . .	40
Compatibility with <code>Lua<math>\TeX</math></code> RTL languages. . . . .	1
Debug <code>\Xonlypstart</code> when using <code>\Xnumberonlyfirstinline</code> and the current line number differs from the previous. . . . .	40
v1.19.0.	
<code>\footssplitskips</code> : <code>\footssplitskips</code> doesn't set <code>\floatingpenalty</code> to <code>\@MM</code> when processing parallel pages. . . . .	166
<code>\xxref</code> : <code>\xxref</code> works also with right side numbers, when <code>\@Rlineflag</code> is not empty. .	269
General: <code>\Xmaxhnotes</code> and <code>\maxhnotesX</code> work now for both two-columns and three-columns setting. . . . .	1
Compatibility with <code>eledpar v1.13.0</code> . . . . .	1
v1.19.1.	
General: Call <code>\correct@footinsX@box</code> and <code>\correct@Xfootins@box</code> directly in <code>\print@notesX@forpages</code> and <code>\print@Xnotes@forpages</code> , that is in <code>eledpar</code> . . .	1
v1.20.0.	
<code>\printlines</code> : Added <code>\ifl@d@Xmorethantwolines</code> and <code>\ifl@d@Xmorethantwolines</code> to <code>\printlines</code> . . . . .	196
<code>\stanza</code> : <code>&amp;</code> and <code>&amp;</code> can be preceded by spaces. . . . .	305
<code>\xxref</code> : Debug <code>\xxref</code> when not loading <code>eledpar</code> (fix bug added in 1.19.0). . . . .	269
General: Add <code>\Xendboxlinenum</code> . . . . .	42
Add <code>\Xtwolines</code> and <code>\Xmorethantwolines</code> hooks . . . . .	39
Add series option. . . . .	1



Correct <code>\Xinplaceofnumber</code> hook. . . . .	1
Explicit error message when calling <code>\Xfootnote</code> outside of <code>\edtext</code> . . . . .	1
Fix bug with line number typesetting direction when using <code>\eledsection</code> and similar commands for RTL texts with <code>Lua<math>\TeX</math></code> . . . . .	1
Fix issues with RTL text in notes when using <code>Lua<math>\TeX</math></code> . . . . .	1
Options fulllines in <code>\Xfootnote</code> . . . . .	25
The <code>\newifs</code> are not followed by boolean values set to false, because it is the <code><math>\TeX</math></code> default setting. . . . .	1
v1.21.0.	
<code>\@edindex@hyperref</code> : Look at the hyperindex option of hyperref before inserting hyperref . . . . .	295
<code>\l@d@section</code> : <code>\endnotes</code> take five arguments. . . . .	223
<code>\ledinnotemark</code> : Add <code>\ledinnotemark</code> . . . . .	292
<code>\ledsetnormalparstuffX</code> : <code>\ledsetnormalparstuff</code> is deprecated and becomes <code>\ledsetnormalparstuffX</code> and <code>\Xledsetnormalparstuff</code> . . . . .	217
<code>\n@num</code> : <code>\n@num@ref</code> deleted . . . . .	112
<code>\n@num</code> defined only one time for both <code>Eledmac</code> and <code>Eledpar</code> . . . . .	112
<code>\newhookcommand@series</code> : <code>\newhookcommand@series</code> can take an optional argument. . . . .	248
<code>\newhooktoggle@series</code> : <code>\newhooktoggle@series</code> can take an optional argument. . . . .	249
<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> . . . .	220
<code>\seriesatend</code> : Fix spurious space in <code>\seriesatend</code> . . . . .	246
<code>\skipnumbering</code> : <code>\skipnumbering</code> defined only one time for both <code>Eledmac</code> and <code>Eledpar</code> . . . . .	123
Correct <code>\skipnumbering</code> for stanza. . . . .	123
Delete <code>\skipnumbering@reg</code> . . . . .	123
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 . . . . .	245
Add <code>\applabel</code> and related . . . . .	57
Add <code>\beforenotesX</code> and <code>\Xbeforenotes</code> features for notes set in two and three column. . . . .	1
Add <code>\hidenumbering</code> . . . . .	23
Add <code>\Xcolalign</code> and <code>\colalignX</code> hooks . . . . .	46
Add <code>\Xendtwolines</code> , <code>\Xendmorethantwolines</code> , <code>\Xendtwolinesbutnotmore</code> and <code>\Xendtwolinesonlyinsamepage</code> . . . . .	40
Add <code>\Xparindent</code> and <code>\hangindentX</code> . . . . .	45
Add <code>\Xtwolinesbutnotmore</code> and <code>\Xtwolinesonlyinsamepage</code> . . . . .	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>\maxnotesX</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>oldprintnpnumspace</code> option allows to come back to previous setting . . . . .	1
<code>parapparatus</code> option works now with familiar footnotes. . . . .	1
Provide <code>\@gobblefive</code> . . . . .	84
v1.22.0.	
<code>\ledinnote</code> : <code>\ledinnote</code> takes a first optional argument, which is the label for hyperlinks. . . . .	292
General: Add <code>\doendnotesbysection</code> command. . . . .	26
Add option for lemma separator inside endnotes . . . . .	43
Adds hyperlink for references to notes in indices. . . . .	1
Fix conflict between <code>noend</code> package option and <code>edtabularx</code> environments . . . . .	1
Provides support for <code>xindy</code> . . . . .	1
Standardize endnotes handbook. . . . .	26
When using <code>hyperref</code> package, internal links in index or with <code>\edlineref</code> are now targeted to the top and not longer to the bottom of the lines they refer to. . . . .	1
v1.22.1.	
<code>\prevpage@num</code> : Correct double symbol when using both <code>\parafootsep</code> and <code>\Xsymlinenum</code> . . . . .	178
General: Fix bug (added on v1.22.0) with <code>\Xinplaceofnumber</code> hook. . . . .	1
v1.23.0.	
<code>\@edtext@level</code> : The boolean <code>\if@edtext@</code> becomes the counter <code>\edtext@level</code> . . . . .	127
<code>\SErefwithpage</code> : Debug <code>\Xendtwolines</code> , <code>\Xendmorethantwolines</code> , <code>\Xendtwolinesbutnotmore</code> and <code>\Xendtwolinesonlyinsamepage</code> when using <code>\apprefwithpage</code> . . . . .	273
<code>\lemma</code> : Fix spurious space after <code>\lemma</code> command . . . . .	131
<code>\newseries@</code> : Prevent spurious spaces when <code>\Afootnote</code> and similar commands are followed by spaces (bug added on 1.0.0). . . . .	236
<code>\sameword</code> : In order to allow use of <code>\sameword</code> with <code>inputenc</code> , we detokenize its mandatory argument before using it in control sequence names. . . . .	135
General: Add <code>\Xboxlinenumalign</code> and <code>\Xendboxlinenumalign</code> . . . . .	42
Add <code>\Xboxstartlinenum</code> , <code>\Xendboxstartlinenum</code> , <code>\Xboxendlinenum</code> , <code>\Xendboxendlinenum</code> . . . . .	42
Allow use of <code>\sameword</code> with <code>inputenc</code> managing of UTF-8. . . . .	1
Compatibility between <code>nofamiliar/nocriticals</code> option and <code>minipage/ledgroup</code> . . . . .	1
Error message when using <code>\beginnumbering... \endnumbering</code> without <code>\pstart</code> . . . . .	1
Fix bug with <code>\sameword</code> when the lemma overlaps multiple line. . . . .	28
Fix bug with <code>\sameword</code> when the same lemma is used for multiple notes or for nested <code>\edtexts</code> . . . . .	28
Fix bug with <code>\skipnumbering</code> called immediately after a <code>\pstart</code> . . . . .	1
Fix error of <code>\iftrue</code> not closed. . . . .	1
Fix spurious space with <code>\skipnumbering</code> (bug added on v1.21.0). . . . .	1
New tools to ensure the line-list file uses the right version of commands when upgrading the <code>eledmac</code> version. . . . .	1
Optional argument of <code>\sameword</code> can be a comma-separated list of <code>\edtext</code> depth. . . . .	28
v1.23.1.	
General: Fix bug with <code>\lemma</code> 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 <code>\AtEveryPstart</code> and <code>\AtEveryPend</code> providing to it an empty argument. . . . .	1
v1.24.1.	
General: <code>\lemma</code> is disabled when using ‘ <code>nocritical</code> ’ option. . . . .	1
v1.24.2.	
General: Fix incompatibility between ‘ <code>nofamiliar</code> ’ option and ‘ <code>memoir</code> ’ 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 <code>\edindex</code> when using <code>xindy+hyperref</code> option. . . . .	1
v1.24.5.	
General: Fix bug of indent, when a added in 1.1.0, when a <code>\beginnumbering</code> immediately follow a sectioning command. . . . .	1
v2.0.0.	
<code>\@iiiminipage</code> : Patch <code>\@iiiminipage</code> instead of redefining it. . . . .	286
<code>\@xympar</code> : Patching <code>\@xympar</code> instead of redefining it . . . . .	276
<code>\endminipage</code> : Patch <code>\endminipage</code> instead of redefining it. . . . .	286
<code>\initnumbering@quote</code> : <code>\initnumbering@sectcmd</code> becomes <code>\initnumbering@quote</code> . . . . .	338
<code>\l@advance@parledgroup@beforenormalnotes</code> : Some conde of <code>\l@dumboxmpfoot</code> moved to <code>\l@advance@parledegroupp@beforenormalnotes</code> . . . . .	287
<code>\newseries@</code> : One endnotes file by series. . . . .	242
General: <code>\@makecol</code> , <code>\@reinserts</code> and <code>\@doclearpage</code> are patched instead of begin redefined . . . . .	258
<code>\doxtrafeeti</code> becomes <code>\Xdo@feet</code> ; <code>\doxtrafeetii</code> becomes <code>\do@Xfeet</code> ; <code>\@opxtrafeeti</code> becomes <code>\@opfeetX</code> ; <code>\doreinxtrafeetii</code> becomes <code>\X@doreinfeet</code> ; <code>\doreinxtrafeeti</code> becomes <code>\@doreinfeetX</code> . . . . .	258
Add <code>\Xendinplaceofnumber</code> hook. . . . .	1
Add <code>\Xendnonumber</code> hook. . . . .	1
Add <code>nonum</code> 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 $\TeX$ ’s output macros are now patched and not override. . . . .	1
Package’s name becomes <code>reledmac</code> . . . . .	1
Patch <code>\@footnotemark</code> instead of redefine it . . . . .	199
Suppress indexing command specific to <code>memoir</code> . . . . .	293
v2.0.1.	
General: Fix bug in <code>eledmac-compat</code> option . . . . .	1
Fix incompatibility between optional argument of <code>\pstart</code> and <code>\numberpstarttrue</code> . . . . .	1
v2.1.0.	
General: Fix bug with <code>\advanceline</code> at the beginning of a <code>\pstart</code> . . . . .	1
Fix bug with <code>\chapter</code> in optional argument of <code>\pstart</code> in parallel typesetting with <code>scrbook</code> . . . . .	1
Fix bug with <code>\eledchapter</code> in parallel typesetting with <code>scrbook</code> . . . . .	1
Fix bug with <code>\setline</code> at the beginning of a <code>\pstart</code> . . . . .	1
Fix spacing bug with <code>\Xhooknote</code> and <code>\hooknoteX</code> when using them to insert text and not to execute code. . . . .	1

New tools to number stanzas . . . . .	1
v2.1.1.	
General: Fix bug with <code>\ledpbsetting{before}</code> . . . . .	1
v2.1.2.	
General: Fix bug with lineation by <code>pstart</code> and <code>tabular</code> environments (added in 2.1.0). . . . .	1
v2.1.3.	
<code>\ledsetnormalparstuffX</code> : Replaced <code>\noindent</code> with <code>\parindent</code> set to 0pt. . . . .	217
General: <code>\Xhangindent</code> and <code>\hangindentX</code> work now with all the paragraphs in the	
note. . . . .	1
<code>\Xnoindent</code> and <code>\noindentX</code> work now again (broken in 2.0.0). . . . .	1
Change some internal code in order to provide compatibility with $\TeX$ release of	
october 2015 . . . . .	1
Fix bug which inserted double space before paragraphed familiar notes. . . . .	1
Fix bug with <code>\edindex</code> when using not-Latin characters without UTF-8 engines . . . . .	1
v2.2.0.	
General: Fix bug with combination of <code>\onehalfspacing</code> and two columns and three	
columns notes typeset. . . . .	1
Fix bug with some setting command and optimization option. . . . .	1
Fix spurious space with paragraphed critical notes when using $\text{Lua}\TeX$ . . . . .	1
Increase line list version number to ensure compatibility with new options of	
<code>reledpar</code> package. . . . .	1
New setting tools for endnotes: <code>\Xendnumberonlyfirstinline</code> ,	
<code>\Xendnumberonlyfirstintwolines</code> , <code>\Xendsymmlinenumber</code> , <code>\Xendbeforenumber</code> ,	
<code>\Xendafternumber</code> , <code>\Xendbeforemsymmlinenumber</code> , <code>\Xendaftersymmlinenumber</code> ,	
<code>\Xendboxsymmlinenumber</code> , <code>\Xendhangindent</code> , <code>\Xendbhooklinenumber</code> ,	
<code>\Xendahooklinenumber</code> , <code>\Xendbhookinplaceofnumber</code> ,	
<code>\Xendahookinplaceofnumber</code> . . . . .	1
v2.2.1.	
General: Compatibility with $\text{TeXformat}$ 2015/10/01. . . . .	1
v2.2.2.	
General: Fix bug in <code>\sethangingsymbol</code> . . . . .	1
Fix bug with old version of <code>etex</code> . . . . .	1
v2.3.0.	
General: Disable empty lines as paragraph in stanza. . . . .	1
Fix compatibility of paragraphed footnotes with <code>bidi</code> v17.9 and following. . . . .	1
Warning message when using some setting commands inside <code>rightside</code> environment	
(deprecated behavior) . . . . .	1
v2.3.1.	
General: Fix spurious space when using optional argument of <code>\stanza</code> (introduced in	
v2.3.0). . . . .	1
v2.4.0.	
<code>\footnoteoptions@</code> : First argument of <code>\footnoteoption@</code> is now mandatory, not	
optional. . . . .	163
General: <code>\Xbhooknote</code> and <code>\bhooknoteX</code> work with notes in columns. . . . .	1
Fix bug of <code>\parindentX</code> and <code>\Xparindent</code> with two columns and three columns	
notes. . . . .	1
Fix bug with <code>\sameword</code> in right side. . . . .	1
Fix spurious space in two columns and three columns notes. . . . .	1
Fix spurious space when using optional argument of <code>stanza</code> (introduced in v2.3.0). . . . .	1
New hooks: <code>\Xlinerangeseparator</code> and <code>\Xendlinerangeseparator</code> . . . . .	39

Option <code>linrangesep</code> for critical footnotes and endnotes. . . . .	39
v2.4.1.	
General: Fix bug with <code>\appref</code> and <code>\apprefwithpage</code> (introduced in v2.4.0). . . . .	1
Fix bug with tabular environments when using <code>babel</code> or <code>polyglossia</code> languages that override $\LaTeX$ <code>\roman</code> command, like Greek language. . . . .	1
Fix bug with tabular environments when using <code>babel</code> or <code>polyglossia</code> languages that override $\LaTeX$ <code>\roman</code> command, like Greek. . . . .	1
v2.5.0.	
<code>\SErefwithpage</code> : Debug <code>\setapprefprefixsingle</code> . . . . .	272
<code>\edlabel</code> : Fix bug when calling <code>\edlabel</code> in a footnotes of the rightside . . . . .	261
<code>\l@d@section</code> : <code>\endnotes</code> take six arguments. . . . .	223
<code>\printlines</code> : <code>\printlines</code> takes an eighth argument: the line flag . . . . .	195
<code>\xlineref</code> : <code>\xlineref</code> does not include anymore the side flag. Use <code>\xflagref</code> to get it. Not that <code>\edlineref</code> still contains the flag. . . . .	267
General: <code>\apprefwithpage</code> and <code>\appref</code> print double quotation mark when the label was not defined. . . . .	1
<code>\apprefwithpage</code> and <code>\appref</code> work with right side crossref. . . . .	1
<code>\apprefwithpage</code> works also when <code>noend</code> option is enabled. . . . .	1
<code>\appref</code> and <code>\apprefwithpage</code> can take <code>linrangesep</code> optional argument. . . . .	1
<code>\edlabel</code> works now in <code>Xfootnote</code> . . . . .	1
<code>\lemma</code> can be used even when the <code>nocritical</code> is enabled. . . . .	1
Compatibility with new hook and tools of <code>reledpar</code> 2.6.0. . . . .	1
Fix spurious vertical space in <code>astanza</code> environment ( <code>reledpar</code> ) . . . . .	1
Log now states ‘There were undefined references’ when using wrong references in <code>\edlineref</code> or <code>edpageref</code> . . . . .	1
New hooks to customize page and line number appearance in endnotes. . . . .	1
New hooks: <code>\Xhookgroup</code> and <code>\bhookgroupX</code> . . . . .	1
New tools to easily make cross-reference to a passage defined by a start and an end line	57
v2.6.0.	
General: Adds compatibility with <code>innnote</code> and <code>notenumber</code> options of <code>indextools</code> package. . . . .	1
Fix bug with footnote counter in <code>ledgroup</code> (added in v2.5.0). . . . .	1
Fix bug, introduced in v2.5.0, with footnote numbering in parallel typesetting when using <code>perpage</code> package. . . . .	1
v2.7.0.	
<code>\@k</code> : <code>\rigidbalance</code> is split in <code>\Xrigidbalance</code> and <code>\rigidbalanceX</code> . . . . .	179
<code>\l@d@section</code> : <code>\endnotes</code> take seven arguments. . . . .	223
General: Add dash as default page range separator for <code>\SEonlypage</code> . . . . .	1
Debug <code>\SErefonlypage</code> when referring to only one page. . . . .	1
Delete parenthesis after <code>\SErefonlypage</code> . . . . .	1
Fix (again) bugs with footnote numbering in parallel typesetting while using <code>ledgroup</code> environments (bug added in v2.5.0). . . . .	1
Fix bug with <code>\SErefwithpage</code> . . . . .	1
Fix bugs in compatibility with <code>innnote</code> and <code>notenumber</code> options of <code>indextools</code> package, when indexing outside of a <code>ledgroup</code> . . . . .	1
New commands to make glossaries connected to page and linenumber with the <code>glossaries</code> package . . . . .	1
New hooks: <code>\Xhsize</code> and <code>\hsizeX</code> . . . . .	49
New hooks: <code>\Xlemmafont</code> and <code>\Xendlemmafont</code> . . . . .	44

New setting commands: <code>\setSErefonlypageprefixsingle</code> and <code>\setSErefonlypageprefixmore</code> . . . . .	1
Warning for duplicate and undefined labels are parsable by latexmk . . . . .	1
Warning for duplicate labels does not send any more a false line and page number . . . . .	1
When using <code>hyperref</code> package, add link in familiar footnotes between the footnote marks in the text and the footnote marks in the footnote . . . . .	1
When using <code>hyperref</code> package, add links for <code>\SEref</code> and <code>related</code> , <code>\appref</code> and <code>related</code> . . . . .	1
When using <code>hyperref</code> package, add links from critical footnotes and critical endnotes to the line of text they refers . . . . .	1
v2.7.1.	
General: Debug <code>\Xhookgroup</code> hooks executed on columnar footnotes (moved to a larger group, to take effect). . . . .	1
v2.7.2.	
General: <code>\Xhsize</code> and <code>\hsizeX</code> become <code>\Xwidth</code> and <code>\widthX</code> . . . . .	49
Fix problem of hyphenation when using <code>hyperref</code> package (added in v2.7.0). . . . .	1
v2.8.0.	
<code>\l@d@section</code> : <code>\Xendhangindent</code> and <code>\Xendafternote</code> can take values which are relative to the font size of the endnote. . . . .	223
General: <code>reledmac</code> cross-referencing can take advantage of <code>xr</code> package. . . . .	1
More <code>\edgls...</code> commands. . . . .	1
No indentation for paragraphed notes in <code>ledgroup</code> . Can be changed with <code>\Xparindent</code> and <code>\parindentX</code> . . . . .	1
v2.8.1.	
General: Warnings for undefined labels are really parsable by latexmk . . . . .	1
v2.8.2.	
General: Fix bug concerning indent in a paragraph immediately following a sectioning command (bug NOT fixed on <code>reledpar</code> ) . . . . .	1
Fix bug with <code>\AtEveryPstart</code> added in version 2.0.0. . . . .	1
Fix bug with vertical space after the between-sectioning command as optional argument of a <code>\pstart</code> and <code>\pstart</code> content . . . . .	1
v2.9.0.	
General: Allow continuing line numbering between normal text and parallel text, using <code>\pausenumbering</code> and <code>\resumenumbering</code> and the <code>continuousnumberingwithcolumns</code> option. . . . .	1
Fix bug when using <code>\linenum{page}</code> and <code>\pausenumbering... \resumenumbering</code> . . . . .	1
Fix bug with three- and two-column footnote setting (added in v2.4.0). . . . .	1
Fix spurious space inside three-column familiar footnote. . . . .	1
Write correct metadata in numbered files when using <code>\pausenumbering... \resumenumbering</code> . . . . .	1
v2.9.1.	
General: Fix bug when notes start with “plus” or “minus”. . . . .	1
v2.9.2.	
General: Fix bug with <code>hyperref</code> package when a lemma starts with “plus” or “minus” (bug introduced in v. 2.7.0). . . . .	1
v2.9.3.	
General: Fix bug with line number position and reset added by v. 2.9.0 . . . . .	1
v2.10.0.	
<code>\print@lemma</code> : Code refactoring between <code>\parafootfmt</code> , <code>\twocolfootfmt</code> , <code>\threecolfootfmt</code> and <code>\normalfootfmt</code> . . . . .	187

General: Add <code>\AtEveryStanza</code> and <code>\AtEveryStopStanza</code> .	1
Fix bug in <code>\ledlsnotefontsetup</code> and <code>\ledrsnotefontsetup</code> which could not handle <code>\color</code> command properly.	1
More specific error messages.	1
New hooks: <code>\Xwrapcontent</code> , <code>\Xendwrapcontent</code> and <code>\wrapcontentX</code> .	45
New hooks: <code>\Xwraplemma</code> and <code>\Xendwraplemma</code> .	44
v2.10.1.	
General: Add ‘nopenalties’ option.	1
Fix bug introduced in v. 1.4: not paragraphed critical footnotes could prevent marginal line number from being displayed	1
v2.11.0.	
<code>\do@actions@fixedcode</code> : Add action 1010	152
General: Add new tools to produce an apparatus of manuscripts	1
Fix bug in <code>\Xparafootsep</code> in parallel typesetting	1
Make <code>\parafootsepX</code> work	1
Prevent <code>\Xtxtbeforenotes</code> hook from causing notes to go beyond the bottom margin	1
v2.12.0.	
General: <code>\preXnotes</code> becomes <code>\Xprenotes</code> (naming convention)	1
Add <code>\hidenumberingonleftpage</code> and <code>\hidenumberingonrightpage</code>	1
Add <code>\toendnotes</code> and related.	1
Add <code>auxdir</code> option.	1
Fix bug in critical and familiar footnotes when using uppercase letters with accent mark	1
Fix bug when using <code>\chapter</code> in optional argument of <code>\pstart</code> in parallel typesetting in combination with the <code>noeledsec</code> option.	1
Fix bug with <code>\ledinnernote</code> and <code>\ledouternote</code> in parallel typesetting	1
Fix bug with familiar footnote number in optional argument of <code>\pstart</code> or <code>\pend</code> in parallel typesetting	1
Fix spurious vertical space in <code>\chapter</code> when used as optional argument of <code>\pstart</code> in parallel typesetting.	1
Make endnote compatible with <code>\sameword</code> mechanism	1
More accurate message to control the position of <code>\Xfootnote</code> and <code>\applabel</code> in the $\TeX$ code	1
v2.13.0.	
General: Version 2.13.0 never existed.	1
v2.13.1.	
General: In critical footnotes, the right side flag is printed only if requested explicitly with <code>\Xlineflag</code> (bug added in v. 2.5.0).	1
v2.13.2.	
General: Fix bug added in v. 11.2 which could make parallel typesetting not work.	1
v2.13.3.	
General: Makes <code>\Xendafterpagenumbe</code> affecting <code>\Serefwithpage</code>	1
v2.14.0.	
General: Hyperref with the line number inside critical footnotes is correct when using <code>\xxref</code>	1
Some internal changes for new features of <code>reledpar</code> .	1
v2.14.1.	
General: Fix bug when using <code>\footnoteX</code> in the first argument of <code>\edtext</code> .	1
v2.14.1a.	
General: Fix problematic typos in the handbook.	1

v2.15.0.	
General: Add ‘byline’ arrangement. . . . .	1
Fix <code>\Xtxtbeforenotes</code> in <code>ledgroup</code> . . . . .	1
v2.15.1.	
General: Fix <code>\edindex</code> in tabular environments. . . . .	1
v2.15.2.	
General: Fix bug with <code>fancyhdr</code> package 3.8 and later. . . . .	1
v2.15.3.	
General: Fix bug with <code>\section</code> in optional argument of <code>\pstart</code> and empty line before <code>\pend</code> (bug added in v2.8.2). . . . .	1
Simplification of the sectioning command code. . . . .	1
v2.16.0.	
General: <code>\Xdo@feet</code> becomes <code>\do@Xfeet</code> . . . . .	258
Add <code>\Xendpagenumberonlyfirst</code> , <code>\Xendpagenumberonlyfirstifsingle</code> , <code>\Xendpagenumberonlyfirstintwo</code> , <code>\Xendinplaceofpagenumber</code> and <code>\Xendsympagenum</code> hooks. . . . .	1
Add <code>\Xpagelinesep</code> hook. . . . .	41
Compatibility with new features of <code>reledpar</code> . . . . .	1
Deleted dead code. . . . .	1
Display a warning message if using a version of $\TeX$ that is too old. . . . .	1
Fix bug with <code>\Xgroupbylines</code> for notes in two columns . . . . .	1
Fix bug with <code>\Xtxtbeforenotes</code> for notes in three or two columns . . . . .	1
Fix bug with ‘notenumber’ option of <code>indextools</code> package when indexing texts in familiar footnotes. . . . .	1
Fix potential bug when using <code>\edindex</code> in critical footnotes. . . . .	1
More explicit error message in case the stanza indentation is not defined. . . . .	258
New options for <code>\fnpos</code> and <code>\mpfnpos</code> to set a customized order for familiar and critical footnotes. . . . .	1
When <code>\edindex</code> is called outside of a <code>\beginnumbering... \endnumbering</code> structure, it is automatically switched to <code>\index</code> , with a warning message. . . . .	1
When indexing texts in familiar footnotes with <code>\edtext</code> , refer to the line number where the footnote is called. . . . .	1
When indexing texts in sidenotes with <code>\edtext</code> , refer to the line number where the sidenote is called. . . . .	1
v2.16.1.	
General: Fix bug with redefinition of the style of the footnote number (bug added in v2.12.0) . . . . .	1
v2.16.2.	
General: Error message if <code>footmisc</code> is loaded after <code>reledmac</code> . . . . .	1
Fix bug introduced by v2.16.1 when using non-expandable control sequence, like <code>\normalfont</code> , in the footnote number style. . . . .	1
v2.16.3.	
General: Fix bug with <code>\Seref</code> (bug added in v2.7.0). . . . .	1
v2.16.4.	
General: Fix bug with vertical space before sectioning command in optional argument of <code>\pstart</code> (bug added in v2.15.3). . . . .	1
v2.16.5.	
General: Fix potential spurious spaces in endnotes. . . . .	1



v2.16.6.	
General: Fix bug with the line number style in <code>\doennotes</code> when referring to right side line in parallel typesetting. . . . .	1
Take into account <code>\linenumberstyle</code> when using <code>\edlineref</code> . . . . .	1
v2.16.7.	
General: Fix bug with <code>\msdata</code> when using multiple <code>\beginnumbering... \endnumbering</code> . . . . .	1
Fix bug with <code>\numberpstarttrue</code> when using multiple <code>\beginnumbering... \endnumbering</code> . . . . .	1
v2.16.8.	
General: Fix bug with <code>\edindex</code> in footnotes in parallel typesetting. . . . .	1
v2.17.0.	
General: Add <code>\edglsadd</code> command. . . . .	1
Add <code>\setmsdataposition</code> setting. . . . .	1
v2.17.1.	
General: Fix spurious space in paragraphed footnotes when using <code>Lua<math>\TeX</math></code> without using Right-To-Left text. . . . .	1
v2.17.2.	
General: Change log message when numbered files still don't exist, in order to improve compatibility with <code>latexmk</code> . . . . .	1
v2.17.3.	
General: Fix bug with <code>\doendnotesbysection</code> and <code>\doendnotes</code> . . . . .	1
v2.17.4.	
General: Fix bug with <code>\setSerefonlypageprefixsingle</code> and <code>\setSerefonlypageprefixmore</code> . . . . .	1
v2.17.5.	
General: Fix bug with <code>\pstartref</code> when referring to the left side in parallel typesetting. . .	1
v2.18.0.	
General: Fix bug when using a <code>\edtext</code> in two lines or more in right-to-left typesetting with <code>X<math>\TeX</math></code> . . . . .	1
Fix bug when using both <code>\Xnumberonlyfirstintwolines</code> or <code>\Xnumberonlyfirstinline</code> and <code>\Xparafootsep</code> and <code>\Xsymlinenum</code> . . . . .	1
v2.18.1.	
General: Fix bug when using <code>\msdata</code> with <code>Lua<math>\TeX</math></code> or with the <code>hyperref</code> package. . . .	1
v2.19.0.	
General: Add <code>\footnoteXmark</code> and <code>\footnoteXtext</code> commands. . . . .	1
Add better compatibility with the <code>csquotes</code> package when using familiar footnotes. . .	1
Fix bug with paragraph indent after sectioning command. . . . .	1
v2.20.0.	
General: Add <code>\AtStartEveryStanza</code> , <code>\BeforeEveryStopStanza</code> , <code>\AtEndEveryPend</code> , <code>\AtStartEveryPstart</code> . . . . .	1
Add second optional argument of <code>\pstart</code> , <code>\pend</code> and <code>\stanza</code> . . . . .	1
Add starred version of <code>\AtEveryPstart</code> , <code>\AtEveryPend</code> , <code>\AtEveryStanza</code> and <code>\AtEveryStopStanza</code> . . . . .	1
Add third and fourth optional argument of <code>\newverse</code> . . . . .	1
Fix bug when using familiar footnotes in <code>\eledsection</code> and related. . . . .	1
Reset font specification at the beginning of familiar footnotes. . . . .	1
v2.21.0.	
General: Add the possibility of nested <code>\sameword</code> . . . . .	1

Fix bug when using formatting command in the argument of <code>\edindex</code> inside <code>\edtext</code> . . . . .	1
Now, as explained in the handbook, an <code>\edindex</code> inside <code>\edtext</code> index only for main text, and not for critical footnotes. . . . .	1